Atom Nieuws

JAARGANG 🌰🌰🌰🌰🌰
nummer 🌰🌰🌰



Bestuur:

Voorzitter:	Secretaris:	Penningmeester:
N. Stad	N. Schoone	M Rutten
Plataanweg 47	Sluispad 55-B	Berkenlaan 24
1544 PB Zaandijk	1521 EZ Wormerveer	3737 RN Groenekan
Tel. 075-280808	Tel. 075-286624	Tel. 03461-3495

Clubwinkel:	Hard-ware cie.:	Redaktie Atom Nieuws:
P. Grevelt	Y. Tuk	H. de Ruiter
Emmastraat 22	Postbus 257	Polarisstraat 25
1782 PD Den Helder	2980 AG Ridderkerk	8303 AC Emmeloord
Tel. 02230-23453	Tel. 01804-20552	

Contributie 1986: fl. 60,00
Leden Buitenland: fl. 75,00

Giro 5244293 Bank 52.84.69.010
Beide t.n.v. Atom Computer Club, Groenekan

Redaktie Atom Nieuws:
Joop Ballijns
Harry de Ruiter
Ed Schijf
Evert van Schothorst
Jaap van de Veen
Gerhard Visser
Wibo Visser

Redaktieadres:
Postbus 1373
8001 BJ Zwolle

Ledenadministratie:	Datasheets:	Drukwerkarchief:
S. van Leeuwen	G. Akkermans	F. Monsanto
Kompasstraat 32	Wikke 1	C. Kohlerstraat 97
1973 PX IJmuiden	1273 BR Huizen	7558 VC Hengelo
Tel. 02550 - 22435	Tel. 02152 - 60294	

Uiterste datum inlevering copy:
nr. 86-4 11-05-1986

DE CLUB-WINKEL:

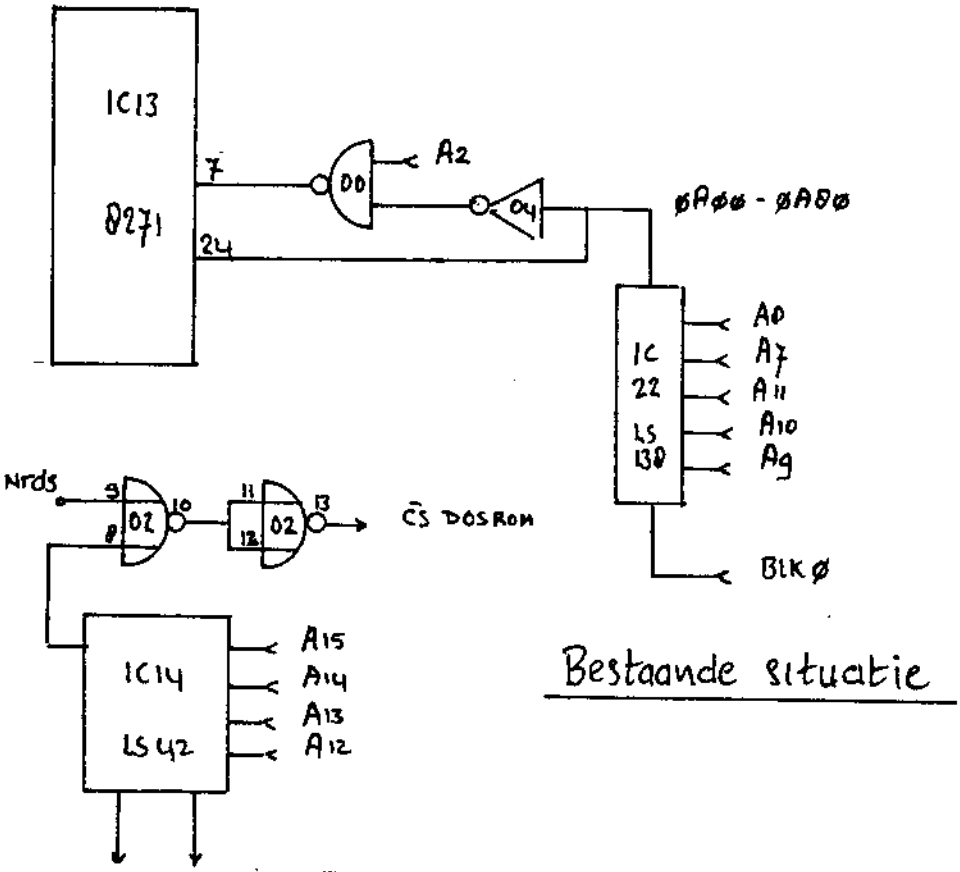
Geheugenkaart; 16 kByte extra in de ATOM	fl. 40,00
Schakelkaart; meerdere EPROM's op Axxx	fl. 47,50
Minischakelkaart	fl. 16,00
Herdruk ACORN NIEUWS 1982; 97 pag. wetenswaardigheden	fl. 6,00
Jaargang 1983; totaal ruim 450 pag.	fl. 30,00
Jaargang 1984; totaal ruim 650 pag.	fl. 35,00
Jaargang 1985; totaal ruim 650 pag.	fl. 35,00
ATOM-WARE 1; Basic interpreter van de ATOM, 98 pag.	fl. 6,00
ATOM-WARE 2; ATOM Disk Operating Systeem, 68 pag.	fl. 5,00
ATOM-WARE 3; ATOM Monitor Operating Systeem, 80 pag.	fl. 5,00

Levering:
Via uw regionale penningmeester of rechtstreeks bij Atom Computerclub te Groenekan, uitsluitend na voorafgaande betaling (vermeerderd met fl. 4,00 voor portokosten).
Indien u 14 dagen na afschrijving van uw bank- of girorekening nog niets hebt ontvangen, informeer dan bij de clubwinkel.

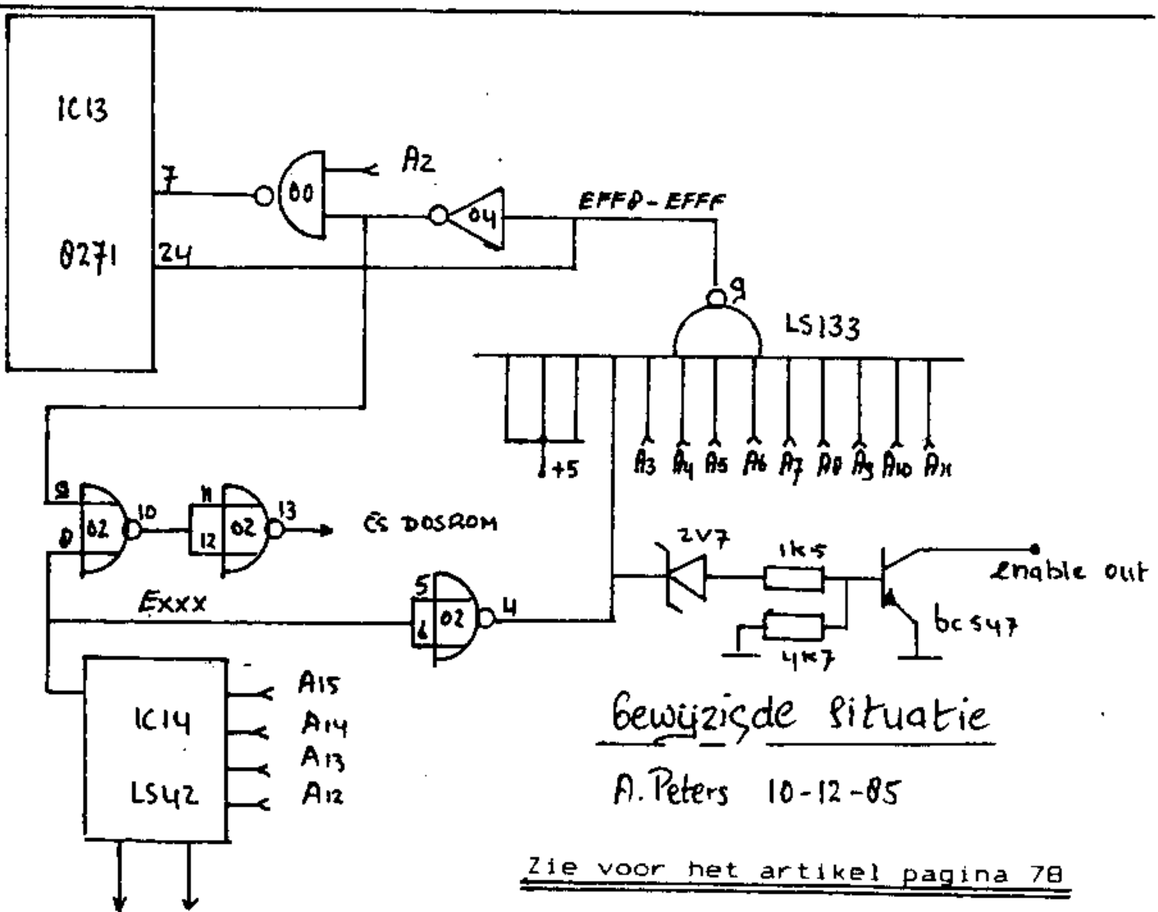
Bij het ter perse gaan van deze uitgave van ATOM NIEUS waren de regioschijven nog niet geheel samengesteld, zodat u de inhoud daarvan niet in dit nummer maar in het volgende nummer van ATOM NIEUWS zult aantreffen.

ATOM NIEUWS is een uitgave van de fed. atom computerclubs n/b
en verschijnt 6 - 8 keer per jaar.

De redaktie gaat er vanuit dat de ingezonden copy gemaakt is door de inzender tenzij uitdrukkelijk anders is vermeld.
De aansprakelijkheid voor auteursrechten voor ingezonden copy ligt dan ook volledig bij de inzender.



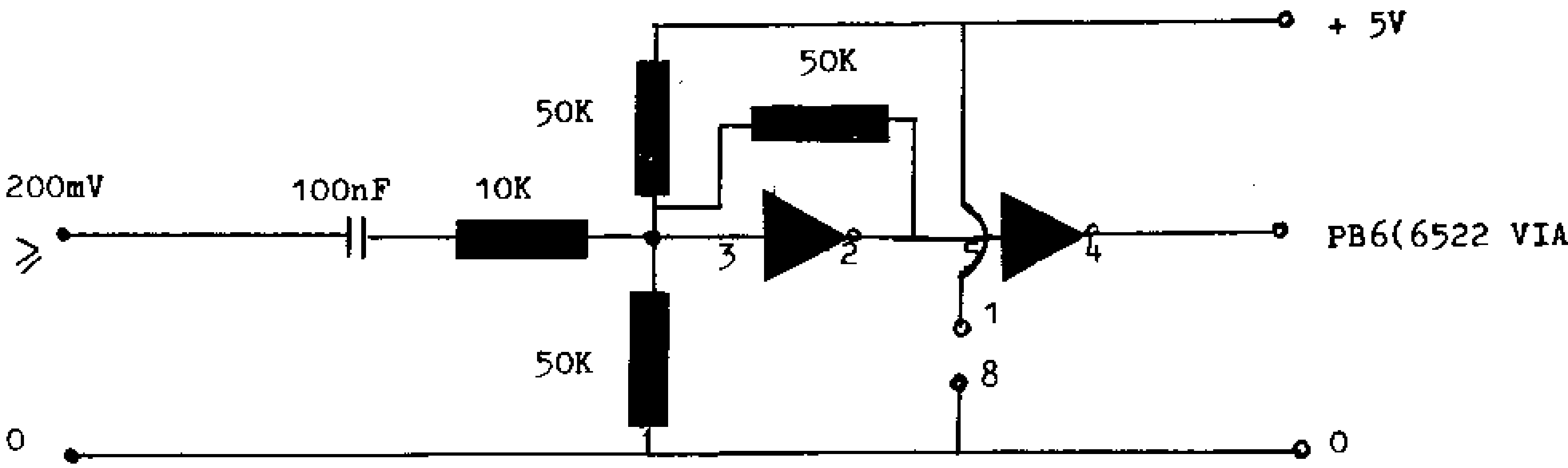
Bestaande situatie



Gewijzigde situatie

A. Peters 10-12-85

Zie voor het artikel pagina 78



IC 1= 1/3 van 4049 Hex inverter buffer

Frequentieberreik 1 - 65535 Hz

De nauwkeurigheid is ± 1 Hz

Overflow	rechtsonder op het scherm	
T1(Via)	midden op het scherm (rechts)	Delay
T2(Via)	boven in het scherm (rechts)	Teller

Het signaal van de te meten frequentie moet via bovenstaande hardware schakeling aangesloten worden op poort PB6 van de VIA 6522

```
  5 CLEARO
 10 F.$21
 15 REM SCAN PROG
 20 DIM LL(4);F.A=1TO2;DIMP(-1);[
 25:LL0 LDA @#0
 26 STA #A0
 27 STA #A1
 30 LDA @#BF
 40 STA #B802
 50 LDA@#60
 60 STA #B80B
 70 LDA@#E0
 80 STA #B80E
 90:LL1 LDX@#0
100 LDA@#FF
110 STA #B804
120 STA #B807
130 STA #B808
140 STA #B809
150:LL2 LDA@#F3 <-- nauwkeurigheid c.q. ijking frequentie teller
160 STA #B805
170:LL3 LDA #B80D
180 AND@#60
190 BEQ LL3
200 ASLA;ASLA;ASLA
220 BCS LL4
230 INX
240 CPX@16
250 BNE LL2
260 LDA #B809
265 EOR@#FF
270 STA #A0
280 STA #801F
290 LDA #B808
295 EOR@#FF
300 STA #A1
310 STA #811F
320 RTS
330:LL4 LDY#EE
340 STY #81FF
350 STY #A2
360 RTS
370];N.;P.$6
380 LINK LL0
390 P.((?#A0*256)+(?#A1))
400 G.10
```

****
PSCREEN-GSCREEN statements door j.bronzwaer
****

De in het verleden al gepubliceerde ideeen over het wegschrijven en terughalen van beeldschermpagina's hebben geleid tot de hier beschreven cocktail die naar mijn smaak een leuke aanvulling betekent voor P-CHARM extra's . Het statement PSCREEN bergt de beeldschermpagina (#8000-#8200) op in een door U te kiezen geheugenpagina tussen #2000 en #A000 (is beveiligd!) . GSCREEN haalt een gekozen geheugenpagina terug naar het scherm in een door U te kiezen mode .
Syntax : PSCREEN#xx,y of GSCREEN#xx,y . xx is de geheugenpagina waar geschreven of teruggehaald moet worden , y is de mode (0-4) waarin dit moet gebeuren .

```
   10 REM **********************
   20 REM * PSCREEN - GSCREEN *
   30 REM **********************
   40
   50 DIMLL12,CC1;F.I=0TO10;LLI=#FFF;N.;CC0=#FFF;CC1=#FFF
   60 P.$12;IN."BEGINADRES TABEL "Y;IN."BEGINADRES  CODE "Z
   70 IFZ>Y AND Z<=(Y+21);P.$7;G.60
   80 P.$21;P=Y+3;GOS.a;P.$6;P=Y+3;GOS.a;@=0
   90 Y?0=#FF;Y?1=#E3;Y?2=#C6;?#3FC=Y/256
  100 P.''"TABEL : #"&Y" - #"&T
  110 P.'"CODE  : #"&Z" - #"&(P-1)';END
  120aT=P;REM tabel
  130 $T="GSCREEN";A=0;GOS.t;$T="PSCREEN";A=1;GOS.t
  140 ?T=#80;P=Z;G.c
  150tT=T+L.T;?T=CCA/256|#80;T?1=CCA%256;T=T+2;R.
  160c
  170[
  180:CC0\code getscreen
  190:LL0 JSRLL11;LDA#92;BNELL1;BEQLL5
  200:LL1 CMP@1;BNELL2;LDA@#30;BNELL5
  210:LL2 CMP@2;BNELL3;LDA@#70;BNELL5
  220:LL3 CMP@3;BNELL4;LDA@#B0;BNELL5
  230:LL4 CMP@4;BNELL8;LDA@#F0
  240:LL5 STA#B000
  250:LL6 DEY;LDA(#90),Y;STA#8000,Y;CPY@#00;BNELL6;INC#91
  260:LL7 DEY;LDA(#90),Y;STA#8100,Y;CPY@#00;BNELL7
  270:LL8 JMP#C55B
  280
  290:CC1\code putscreen
  300 JSRLL11
  310:LL9 DEY;LDA#8000,Y;STA(#90),Y;CPY@#00;BNELL9;INC#91
  320:LL10DEY;LDA#8100,Y;STA(#90),Y;CPY@#00;BNELL10;JMP#C55B
  330
  340:LL11\interpreteren
  350 JSR#C78B;JSR#C231;JSR#C4E1;LDA@0:STA#90;TAY:TAX:STX4
  360 LDA#16;STA#91;CMP@#20;BMILL12;CMP@#9F;BPLLL12
  370 LDA#25;BNELL12;LDA#34:BNELL12;LDA#43;BNELL12
  380 LDA#17;STA#92;RTS
  390:LL12JMP#FA7D
  400]
  410 RETURN
```

ENVELOP

In het kader van het volledig vervangen van de typemachine door de printer heb ik dit programma geschreven. Het maakt het mogelijk om enveloppen van de naam en het adres van de geadresseerde, afzender en eventueel de aanduiding "DRUKWERK" te voorzien. Het programma gebruikt (afgezien van de gedimensioneerde ruimte) geheugen van #2900 tot #333F. Tevens is een P-Charme en printer benodigd.

Het menu is simpel: allereerst wordt gevraagd of de aanduiding "DRUKWERK" in de linkerbovenhoek geplaatst moet worden. Zo ja dan "J", zo nee dan "N" intikken. Wilt u stoppen, dan dient u de "E" van einde in te toetsen. Vervolgens wordt de vraag gesteld of er een afzender op de envelop vermeld moet worden. "J" en "N" spreken voor zich, doch middels het intikken van "Z" geeft u te kennen dat de afzender vermeld moet worden en dezelfde is als de de voorgaande keer opgegeven afzender. Het programma gaat er overigens vanuit dat de invoer correct is geschied!

De lay-out van de envelop is als volgt: indien nodig wordt "DRUKWERK" enlarged, linksboven afgedrukt (en de eventuele afzender linksonder). Indien dat niet gewenst wordt doch een afzender wel, dan verschijnt deze linksboven. De geadresseerde verschijnt altijd op de normale plaats.

De gebruikte printercodes (in de regels 530, 550, 560, 600, 640 en 650) zijn die van de STAR gemini-10X. Voor verandering van deze codes i.v.m. het gebruik van een andere printer verwijs ik naar het overzicht op pag. 84 e.v. van AN 4.2

```
overzicht variabelengebruik (in hoofdprogramma)
A   : codebytes - ?A  = geadresseerde ? J/N/E
                  A?1 = aantal regels geadresseerde
                  A?2 = aantal regels afzender
                  A?3 = afzender ?  J/N/Z
B   : stringpointer             C,F : stringvariabelen
D,E : invoervariabelen (voor de keuzes in het menu)
G,H : booleans                  I,J : tellers in de loops
```

N.B. Het verdient aanbeveling het programma eerst eens met gewoon papier in de printer uit te proberen. Tevens dient men er bij het printen van enveloppen rekening mee te houden dat deze dikker zijn dan gewoon papier.

Peter van Alphen

Het verplaatsen van de Floating-Point opslagruimte.

In de standaard versie van de floating-point ROM worden de floating-point variabelen en de adressen van de F.P.-array's opgeslagen in het geheugengebied #2800 - #28FF.
Voor wie geen floppy disk heeft, maar wel geheugenuitbreiding beneden #2800, zitten de F.P.-variabelen behoorlijk in de weg. Gelukkig is er niet zoveel voor nodig om dit te veranderen. Ik heb mijn F.P.-opslag verplaatst naar #B400 - #B4FF, waar bij mij een stukje extra RAM zit (let op: dit is niet volgens de standaard).
Het is mogelijk de F.P.-opslag te verplaatsen naar ieder gewenst stuk vrije ram. Wil je de F.P.-opslag in het geheugengebied #XX00 - #XXFF, dan moet je de inhoud van de volgende adressen in de F.P.-ROM wijzigen als volgt:

adres (F.P.)	was	wordt
D40D	28	XX
D416	28	XX
D452	28	XX
D485	28	XX
D48C	28	XX

P-charme gebruikt ook een aantal adressen in het gebied #2800 - #28FF. Als je die mee wilt laten verhuizen, dan moet je de inhoud van de volgende adressen in de P-charme wijzigen:

adres (P-charme)	was	wordt
A540	28	XX
A543	28	XX
A5DA	28	XX
A5DF	28	XX
A976	28	XX
ACEB	28	XX
ACEE	28	XX

Wie RAM-geheugen heeft vanaf #0400, zou ook willen dat P-charme eventuele programma's in dit gebied herkent. Helaas is dit niet zo in de huidige versie.
De onderstaande wijzigingen lossen dit op:

adres (P-charme)	was	wordt
A06F	0F	03
A077	A0	F0

Alle programma's die staan in het geheugengebied #0400 - #EFFF worden nu herkend (CAT) en automatisch gestart indien je er om vraagt.

VEEL SUCCES!!!

*** CMOS-ASM SUPPLEMENT ***

VERBETERING 1

Bij de in AN 7 1985 gepubliceerde CMOS-assembler heb ik helaas een kleinigheidje over het hoofd gezien. Indien men een instructie met oude mnemonic maar met nieuwe adresseringsvorm achter een label plaatst, gaat de assembler de mist in.
Dus 160:label ORA (zp) gaat mis.
Dit probleem is overigens op te vangen door een puntkomma ertussen te zetten.
Dus 160:label;ORA (zp) gaat goed.
Toch zie ik dit als een tekortkoming.
Daarom stel ik voor de volgende wijzingen aan te brengen:

```
700 LDY @0
702:TRY'AGAIN
704 LDX @3
706 STY #03
708:NXTMNECHR
710 JSR getchar
712 CMP @CH":"
714 BNE NO'LABEL
716:NXTLABCHR
718 INY
720 LDA (#05),Y
722 CMP @CH" "
724 BEQ TRY'AGAIN
726 CMP @CH";"
728 BEQ ERROR
730 CMP @#0D
732 BNE NXTLABCHR
734 BEQ ERROR
740:NO'LABEL

935:ERROR
```

VERBETERING 2

De BASIC/ASSEMBLER-test is overbodig, terwijl de STY #67 op regel 630 constructief gezien niet op de juiste plaats staat. Daarom:

```
520 CPY @208
530 BEQ TRYNEWMNE
540 CPY @156
550 BEQ TRYNEWADM
695 STY #67
835 LDY #67
855 STY #67
```

DELETE: 560, 570 en 620 t/m 680

ASSEMBLER-MODE DIRECTIVE

```
SYNTAX:  .AM 6502
         .AM R65C02
```

Hiermee kan men de CMOS-assembler in 6502 mode of R65C02 mode zetten.
Op deze manier loopt men niet het risico dat men per ongeluk een CMOS-instructie gebruikt in een voor de 6502 bedoeld programma. Bovendien kan iedereen nu direct zien voor welke processor het programma geschreven is. Weglaten van dit directive geeft 6502 mode.

```
 145 D=1    ;REM AANTAL DIRECTIVES
 150 N=18+D ;REM TABEL-LENGTE
 160 K=10   ;REM INTERNE REF.
 180 DIM B(N*3)
 260 PRINT "DIMTOP"&?35+?36*256'
2871 P?18=MODE%256
2951 P?18=MODE/256
3030 $B+48="BBRBBS.AM"

1041 CPX @N-D
1042 BCC SKPNEWMNE
1043 STY #67
1044 JMP (#0066)
1061 BIT #28FF
1062 BVC ERROR
1063 SEC

2721:MODE
2722 JSR getchar
2723 DEC #03
2724 TAX
2725 LDA #28FF
2726 AND @#BF
2727 CPX @CH"R"
2728 BNE SKIPORA
2729 ORA @#40
2730:SKIPORA
2731 STA #28FF
2732 JMP TERMINATE
```

Na alle wijzigingen op deze bladzijde krijgen we:

```
 10 PROGRAM CMOS-ASSEMBLER 2.0
300 XIF S=#E700 ...
```

TS	2900
TOP	3B32
DIMTOP	3B6C
ENTRY	7300
END+1	74EE

ZELF GRAPHICS ONTWERPEN

Is het u ook al eens overkomen? U ziet uw computer een prachtige grafische tekening maken; als u dan het programma gaat bekijken, in de hoop te ontdekken hoe deze tekening ontstaat, dan zakt de moed u onmiddelijk in de schoenen. Het programma wemelt van de sinussen en cosinussen en moedeloos stopt u ermee. De volgende keer tikt u het programma klakkeloos over, u bewondert de tekening en misschien verandert u op goed geluk af enige berekeningen zonder te weten wat het effect zal zijn. Het gevoel van onvoldaanheid blijft echter bestaan. De bedoeling van dit artikel is om enige tipjes van de sluier die rond deze graphics hangt op te lichten en u aan te zetten tot het zelf creeren van graphics.
De eerste stap die we hiertoe moeten nemen(tevens de moeilijkste) is de verklaring van de woorden sinus en cosinus en aanverwante functies. Hiertoe bekijken we fig.1

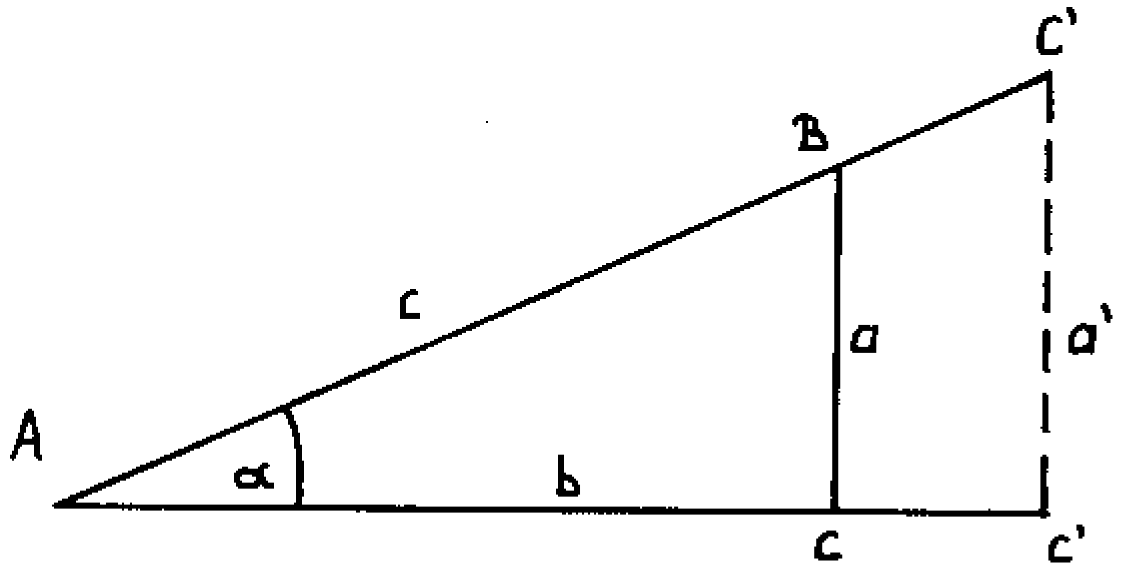


fig. 1

Hier zien we de rechthoekige driehoek ABC. In deze driehoek bestaat er een verband tussen de hoeken en de zijden. Als we het lijnstuk BC delen door het lijnstuk AB of we delen het lijnstuk B'C' door het lijnstuk AC'; de uitkomst blijft hetzelfde. De waarde van de uitkomst wordt dus bepaald door de waarde van de hoek en niet door de waarde van de betreffende lijnstukken. Verandert de waarde van de hoek dan verandert dan verandert eveneens de verhoudingswaarde. Voor de duidelijkheid werden aan deze verhoudingen namen toegekend.
Zo is;

$\frac{BC}{AB}$ = de sinus van hoek α of sin a

$\frac{AC}{AB}$ = de cosinus van hoek α of cos a

$\frac{BC}{AC}$ = de tangens van hoek α of tan a

$\frac{AC}{BC}$ = de cotangens van hoek α of cotg a

Hieruit zijn enige afleidingen te maken. Zo is BC = AB * sin a. en AC = AB * cos a. Bekijk dit even , we komen hier later nog op terug. Het grote nut van deze verhoudingen is dat als een van dze verhoudingen bekend is alle andere te brekenen zijn. Door het bestudern van deze onderlinge betrokkenheid zijn vele andere formules ontstaan. We beperken ons slechts tot enkele algemeen beknde.
Te weten;

$$\tan a * \cot g\ a = 1$$

$$\sin^2 a + \cos^2 a = 1$$

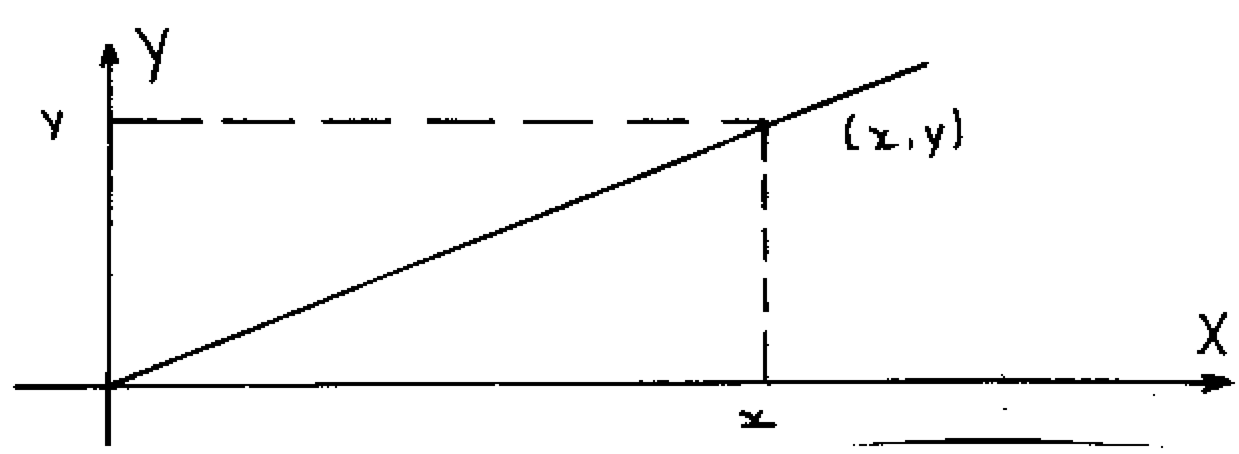
$$\tan a = \frac{\sin a}{\cos a}$$

$$\cot g\ a = \frac{\cos a}{\sin a}$$

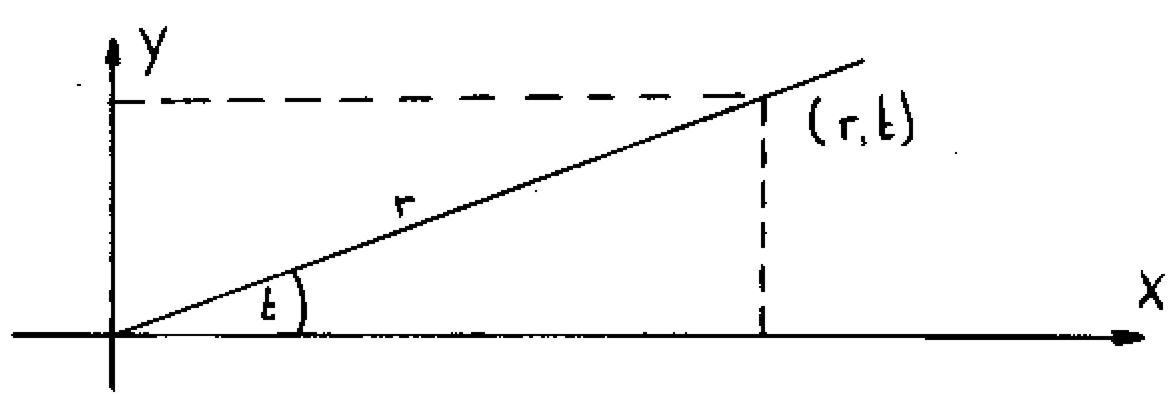
Men heeft tabellen samengesteld waarin deze waarden van de sinus en de cosinus zin opgenomen. Men heeft tafels waarin de waarden zijn opgegeven in vier, soms zelfs in vijf decimalen achter de komma. Voor ons hoeft dit gelukkig niet. Als onze ATOM met een F.P. rom is uitgerust dan zoekt hij deze waarden zelf wel op indin nodig.(opm. cotangens kent hij niet. Maar ja, cos a/sin a = cotg a. Dus geen probleem). Merk op dat, indien u deze tabellen zou bestuderen (u kunt ze plotten op het scherm (manual pag.166)), de waarden van sinus en tangens tussen 0° en 45° stijgend zijn en deze voor cosinus en cotangens dalend.

grafische voorstelling

Een punt in een vlak kan op 2 manieren beaald worden.
Een eerste mothode is de cartesische methode waarin men werkt met rechthoekige coordinaten.(x,y)
voorbeeld;



Een tweede methode is die met poolcoordinaten. (r,t)
voorbeeld;



't' is het maatgetal van de hoek tussen de X-as en de dragende vctorrechte door het gewenste punt. Het opgeven van de hoek t en de waarde van r is voldoende om een punt in een vlak ondubbelzinnig te bepalen
We zullen dit aantonen aan de hand van een programma. Hierin wordt u gevraagd om een hoekwaarde en de lengte van een straal in te geven. Hierna zal het programma het desbetreffende lijnstuk tekenen.
Programma 1

```
  1  P.#12
 10 M=128;N=96
 20 FIN."HOEK"%A
 30 FIN."STRAAL"%R
 40 CLEAR 4
 50 PLOT 4,M,N
 60 %X=%R*COSRAD%A
 70 %Y=%R*SINRAD%A
 80 PLOT 5,(M+%X),(X+%Y)
 90 FOR X=1 TO 120;WAIT;NEXT
100GOTO 1
```

Misschien vraagt u zich af waarom in de formule de functie RAD wordt gebruikt? Ziehier het antwoord. Een hoek kan men op twee manieren uitdruiken nl in graden en in radialen. Daarom moet men op voorhand bepalen welke van de tweee uitdrukkingen men gebruiken wil. Onze ATOM werkt met radialen voor de omzetting in sinus of cosinus of tangens. Geeft men hem de formule sin 90, dan zal hij de waarde 90 beschouwen als zijnde de waarde van een hoek uitgedrukt in radialen. Geeft men hem de formule sinrad 90 dan weet hij dat de waarde van de hoek uitgedrukt is in graden. Deze waarde wordt omgezet in radialen waar dan de sinus van berekend wordt. Onthoudt nog dit. 360° = 2 * PI (RAD).
Als u dit programma enige tijd gerund hebt zal het u wel opgevallen zin dat we in feite de stralen van een cirkel tekenen. Het programma om een cirkel te tekenen is dan ook niet moeilijk meer.
Hier is het;
Programma 2 (cirkel)

```
  1  P.#12
 10 M=128;N=96
 20 FIN."STRAAL"%R
 30 CLEAR 4
 40 F.S=0 TO 360
 50 PLOT 4,M,N
 60 %X=%R
 70 %Y=%R*SINRAD S
 80 PLOT 5,(M+%X),(N+%Y)
 90 NEXT S
100END
```

Veranderen we in dit programma in lijn 80 PLOT 5 door PLOT 13 dan hebben we de cirkelomtrek. Een mooi effect kunnen we bereiken door PLOT 5 te vevangen door PLOT 6. Neem als straal bijvoorbeeld 90.
U zult verbaasd zijn over het resultaat. Door in lijn 40 een step toe te voegen krijgen we een stereffect. Ook kunt u de cirkel op een andere manier laten aangroeien door in lijn 60 en 70 de functie RAD weg te laten. Probeert u het maar even.
Aan deze werkwijze kleeft een nadeel, we kunnen namelijk niet meer dan 360 lijnen of punten bepalen. Willen we meer punten bepalen dan dienen we op een andere manier te werk te gaan.
U weet nog wel dat 360 = 2 * PI (rad.). Indien we deze waarde delen door het aantal punten wat we nodig hebben, dan hebben we de mate van aangroei van de hoek.
Programma 2a

```
  1  P.$12
 10  M=128;N=96
 20  FIN."STRAAL "%R
 30  IN. "AANTAL PUNTEN " S
 40  %S=2*PI/S
 50  %T=0
 60  CLEAR 4
 70  FOR T=1 TO S
 80  PLOT 4,N,M
 90  %T=%T+%S
100 %X=%R*COS%T
110 %Y=%R*SIN%T
120 PLOT 6,(M+%X),(N+%Y)
130 NEXT T
140 END
```

Natuurlijk kan men dit ook oplossen door in lijn 40, 360 te delen door S en in lijn 110 en 110 de functie RAD al dan niet toe te voegen. De hierboven beschreven manier, namelijk de uitdrukking in 2*PI is meer gebruikelijk en sneller in uitvoering.
Suggestie; probeer eens straal 90 en aantal punten 900.
U zult al wel begrepen hebben dat de cirkel bepaald wordt door de formules in lijn 100 en 110. Deze noemt men de parameters.
Kort gezegd de parametervoorstelling van een cirkel is;

x = r * cos t
y = r * sin t

Met deze parameters kan men natuurlijk gaan spelen. Doe dit als oefening.
Probeer de volgende parameters;

1. x = r * cos t ; y = 2 * r * sin t

2. x = r * cos t ; y = r * sin * 2 * t

3. x = r * cos * 3t ; y = r * sin * 2 * t

4. x = r * cos (s+t); y = r * sin * 4 * t

Om het verloop van deze tekeningen beter te kunnen volgen kunt u in lijn 120 PLOT 6 beter vervangen door PLOT 13.

HET " STILZETTEN " VAN HET TV/ MONITORBEELD.

Wanneer een gewone TV gebruikt wordt als monitor voor de ATOM dan blijkt heel vaak dat het beeld staat te "trillen" ondanks een perfecte afstelling van het toestel.
Voor dat trillen zijn een aantal oorzaken te noemen welke ik in dit artikel graag op een rijtje wil zetten en wat nog belangrijker is wat u er aan kunt doen.
Ten eerste waardoor kan het beeld gaan trillen ;
a) ten gevolge van rimpelspanningen op de 5V voedingsspanning
b) ten gevolge van strooivelden van voedingstransformatoren
c) door het verschil in rasterfrequentie van de atom (60 Hz) en de normale rasterfrequentie (50 Hz) waarop de meeste TV toestellen zijn gebouwd en afgeregeld.
d) door een of andere reden die ik niet heb genoemd.

a) De gevolgen van te grote rimpelspanningen voor het beeld.

De aanwezigheid van een rimpelspanning kan het beste worden bekeken met behulp van een oscilloscoop.
Voor de werking van de computer zelf is het doorgaans helemaal geen ramp als er een rimpelspanning van 100mV t-t op de 5V voedingsspanning staat omdat de computer een digitale schakeling is welke maar 2 toestanden kent n.l een"1" (5V) en een "0" (0 volt).
Het onderdeel in de computer welke het videobeeld voor de TV of monitior verzorgt heeft echter wel last van die voedingsrimpel.
Deze schakeling (de videocontroller) moet vanuit een digitale code een analoog signaal produceren waaruit het beeld wordt opgebouwd.
Dat analoge signaal is in stapjes verdeeld van 0-1 volt zodat de helderheid van elk beeldpuntje kan varieren van zwart naar wit.
Wanneer de rimpel op de voedingsspanning nu groter of gelijk is aan zo'n stapje in de helderheidsinformatie dan zal dat direkt te zien zijn op het TV scherm in de vorm van snelle variaties in de helderheid of het trillen van het beeld .

Wat doen we er aan?
1) controleer de rimpelspanning ,deze mag in belaste toestand (dus met de atom aangesloten!!!!) niet groter zijn dan 20 mV top-top (waarjan overigens elke goede gestabiliseerde voeding voldoet!)Controleer ook of de voeding staat te oscilleren , dit kan worden veroorzaakt door het onjuist bedraden van zelfgebouwde regulatorvoedingen.
Een maatregel tegen dat oscilleren is het plaatsen van tantaal elco'tjes (tussen ingang en massa en een tussen uitgang en massa)
Zoals voor elk apparaat geldt, is ook hier een goede voeding van groot belang

b) De gevolgen van strooivelden van nettrafo's

Wanneer men een voedingstransformator te dicht bij de beeldbuis van een TV plaatst waarop men het beeld van een atom weergeeft, dan zullen de elektromagnetische strooivelden vanuit de trafo de afbuiging van de elektronenstraal in de beeldbuis beinvloeden.
Bij een normale TV is dat niet zo'n groot probleem omdat de beeldfrequentie gelijk is aan de netfrequentie (50 Hz) .
Omdat de atom een hogere beeldfrequentie heeft als de netfrequentie (60Hz) zal de beinvloeding van het strooiveld een trilling van het beeld teweeg brengen van 10 Hz.

We kunnen hier een aantal maatregelen tegen nemen:

1) Het verplaatsen van de nettrafo van de TV of monitor zodanig dat deze zover mogelijk is verwijderd van de beeldbuis.
2) De TV via een externe voedingstrafo voeden (trafo buiten de kast van de TV
3) Het afschermen van de voedingstrafo met behulp van mu-metaal.Deze methode wordt veel gebruikt in fabrieksapparaten en geeft en zeer goed resultaat.
4) Het voeden van de monitor met gelijkspanning b.v. 12-15 Volt (dit geldt voor portabele TV's , de spanning moet afgevlakt zijn maar stabilisatie is niet noodzakelijk)

c) Verschil in beeldfrequentie van de ATOM en een normaal TV beeld.

Hieraan kunnen we weinig doen, de beeldfrequentie die ook wel rasterfrequentie wordt genoemd is vastgelegd door het gebruik van de 6847 videoprocessor met het daarbij behorende kristal voor de klokfrequentie.
Via deling vanuit de klokfrequentie van de videoprocessor komen we aan de 60 Hz beeldfrequentie die overigens in de V.S een standaard is (NTSC)
Een aparte videokaart met een andere videoprocessor zoals b.v. de 6845 of de veel modernere broertjes daarvan ,welke een 50 Hz beeld produceren kunnen een oplossing zijn.
Met een 80 Kolommen-kaart die van een dergelijke videoprocessor is voorzien vangen we dus twee vliegen in een klap.

CONCLUSIE: er zijn dus nogal wat oorzaken voor een trillend beeld aan te wijzen !
Meestal is de belangrijkste oorzaak het strooiveld van een of meerdere voedinstrafo's die gemakkelijk te verhelpen zijn door de trafo te verdraaien of te verplaatsen zodat het effect niet meer storend is.
SUCCES!

```
 10 REM 3D GRAPHIC
 20 %Z=90;CLEAR4;F.N=0TO180
 30 %S=2*RADN;%R=%Z*SIN(%S*2)
 40 X=%(%R*SIN%S+128);Y=%(%R*COS%S+96)
 50 IF N=0 THEN MOVE X,Y
 60 DRAW X,Y;%R=%Z*COS(%S*2)
 70 X=%(%R*COS%S+128);Y=%(%R*SIN%S+96)
 80 MOVE X,Y;N.;F.N=0TO200;WAIT;N.;Z=50;CL
100 %S=2*RADN;%R=Z*SIN(%S*2)
110 X=%(%R*COS%S+100);Y=%(%R*SIN%S+140)
120 IF N=0 THEN MOVE X,Y
130 DRAW X,Y;%R=Z*(SIN%S*2-COS%S*2)
140 X=%(%R*SIN%S+100);Y=%(%R*COS%S+140)
150 MOVE X,Y;N.;F.N=0TO200;WAIT;N.;E.
```

Heeft u ook vaker de wens gekoesterd om diverse gepubliceerde , zelfverzonnen of zich in diverse utility's bevindende mooie statements voor Uw al of niet nog te vervaardigen basicprogramma's te verzamelen en door P-CHARM te laten herkennen ? Zoniet , dan komt dat nog wel , zoja dan is onderstaande verzameling misschien iets voor U en wordt U dan ook bij deze hoffelijk aangeboden .
Bij het bestuderen van vele al eerder door harde werkers vervaarigde boxen (goed voor Uw ontwikkeling als assemblerprogrammeur!) heb ik naar statements gezocht die voor basicprogramma's waardevol kunnen zijn maar die eigenlijk onbereikbaar waren omdat mijn eerste voorkeur nu eenmaal uitgaat naar het gebruiken van uitsluitend P-CHARM en ik absoluut geen voorstander ben van het gebruik van schakelsoft binnen basicprogramma's uit oogpunt van uitwisselbaarheid .
Het resultaat is een verzameling van mooie statements , deels letterlijk overgenomen , al of niet gewijzigd of zelfs gerepareerd . Hier komen ze :

BLOCK - CUBE - CIRCLE - NOSNOW - SNOW
----

bron: Gagsrom , zie handleiding A.N.4-6 pag.28

CPC

(Clear Post Cursor) , maakt het scherm schoon (clear0 mode) achter de cursor tot einde scherm.

CURSOR
------

bron: F.le Blanc A.N.4-2 pag.74 . Syntax: CURSORx,y .

ENTER
-----

bron: Charles de Moor . Een nieuw input-statement voor variabelen , uitsluitend bestaande uit gehele of gebroken getallen (E-notatie NIET toegestaan) met een door U op te geven lengte-begrenzing . Syntax : ENTERx , waarin x het maximaal aantal gewenste karakters . Er verschijnt een vraagteken op het scherm , uitsluitend cijfers , slechts 1 minteken en slechts 1 punt worden geaccepteerd en tevens op hun juiste plaatsing beoordeeld . Deleten is normaal mogelijk en het vraagteken blijft staan . Na het intypen van x karakters volgt een automatische return (kortere input met return wordt ook geaccepteerd) en de ingetypte string wordt vervolgens in het inputstringbuffer geplaatst . De instructie %A=VAL$#140 , bijvoorbeeld , plaatst het ingetypte vervolgens in de variabele %A. Gewoon uitproberen , het statement is erg vriendelijk.

FIX
---

bron: cx-2.7h utility van het cx-systeem , een perfecte afrondingsroutine , zie handleiding A.N.4-4 pag 78-79

FLOW-UNFLOW
----

bron: het fantastische BACKUP-programma van Ronald Boers , ge-relocated door Yvo Tuk , zie diens beschrijving in A.N.4-4 pag.6

GINVERT

(Graphic Invert) , een snelle inverteerroutine voor CLEAR4.
Syntax: GINVERT#xxxx,b,l . #xxxx=beginadres op het scherm in clear4 (#8000-#9800) , b = breedte (0-31) , l = aantal lijnen naar beneden (0-192) . De routine werkt ruisvrij en is beveiligd voor inverteren in verboden gebieden.

GRMOD-TXMOD
----

bron: AXR-1 of(?) JOSBOX , gepubliceerd in A.N.3-5 pag.32-34.

PSCREEN-GSCREEN

(Putscreen - Getscreen) bron: ASBK-i box , in nog meerdere vormen gepubliceerd , naar mijn smaak echter nu met de meest universele syntax: PSCREEN#xx (#20....#7E) schrijft het scherm naar de gewenste pagina #xx , GSCREEN#xx haalt het scherm weer terug . Tussen #2000 en #A000 kunnen zodoende 64 tekstpagina's worden opgeborgen.

HEADER
------

bron: DISATOM , echter nu foutloos . Syntax: HEADERx waarbij x een getal tussen 0 en max.6 . Het scherm scrollt vanaf regel x . Uitschakelen van de routine gebeurt met HEADER0.

INVERT
------

bron: FLASH van F.le Blanc , A.N.3-4 pag.61 echter met wijzigingen. Syntax: FLASHx,y,z . x=karakternummer op het scherm (0-511) , y=breedte (tot rand scherm) , z=aantal regels (tot onderzijde scherm . Er wordt slechts eenmaal en uitsluitend tekst geinverteerd , het "flashen" maakt U maar zelf met bijv.: DOKEYK;FLASH;UNTILK<>0 . De routine werkt zeer snel en is sneeuwvrij gemaakt .

SHAPE
-----

bron: het artikel "SHAPES" van Eric Snelders , zie handleiding in A.N.3-7 pag.30-34 waarin U ook de listing van het DESIGN-programma vindt voor deze shapes.

SPEED:

Vertragingsroutine , handig voor het printen van teksten die zorgvuldig moeten worden gelezen . Syntax: SPEEDx , waarin x de met het gekozen getal oplopende vertraging bepaalt . Met SPEED0 wordt de routine uitgeschakeld. (N.B. SPEED , GRMOD en HEADER kunnen niet tegelijk worden gebruikt vanwege het verzetten van vectoren!)

SPLOT
-----

bron: Een cadeautje van Lex van Zandvoort , hij zal dit nog wel publiceren , neem ik aan. Syntax: SPLOTxxxxxxxx , waarin x uitsluitend 1 of 0 mag zijn. Deze routine maakt het plotten van zelfgekozen soorten stippellijnen mogelijk , bijvoorbeeld SPLOT01010101 of SPLOT00110011 . De instructie SPLOT11111111 brengt de plotroutine weer in zijn oorspronkelijke functie terug.

VLIST
-----

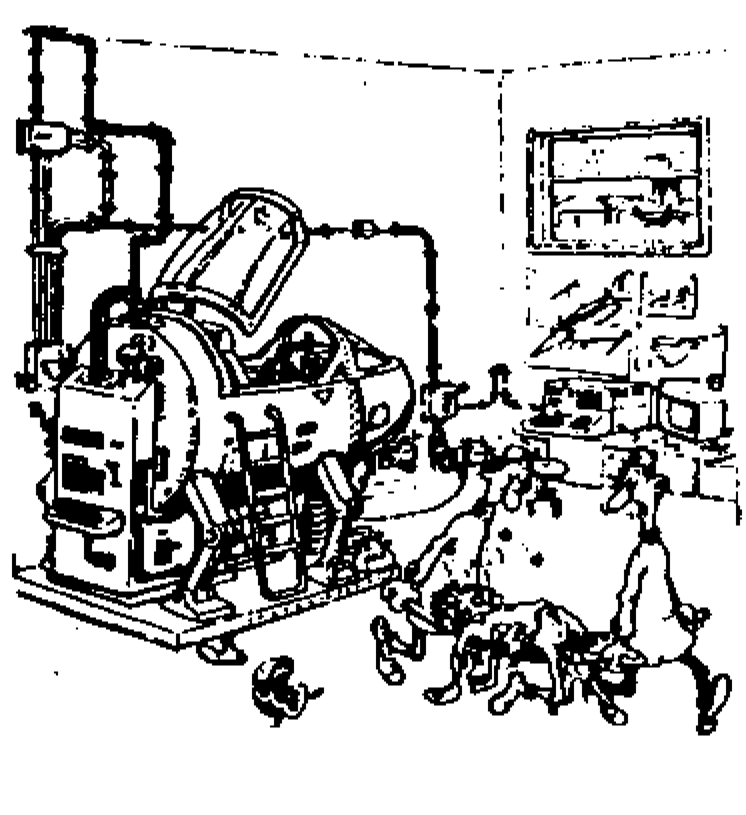
bron: F.v.Hoesel , A.N.3-2 , mei 1984 , pag.50 . De routine is aangepast om probleemloos te werken met Branquart-schakelsoft en de scherm-layout is iets verfraaid. VLIST geeft U op het scherm een overzicht van alle hier genoemde statements met hun beginadres .

Tenslotte:

Dank aan de auteurs van P-CHARM en de diverse routines .
De source van dit geheel is in het listingsarchief opgenomen , zodat U naar wens wijzigingen kunt aanbrengen. De source is ca. 16 kByte lang dus Uw extra statements zullen wel op #7000-#7FFF komen te staan . Daarop is de source dan ook gemaakt en deze "utility" gaf ik dus de naam "70-UTIL" . U vindt de voor #7000 geassembleerde versie onder die benaming samen met de source "70UTILs" op de regioschijf en in het bandjesarchief. Intypen kan ik U niet aanraden. Voor gebruik wel eerst nog even ?#3FC=#70 intypen . Veel plezier met deze verzameling .

vlist PAGE #70

BLOCK	7C76	CIRCLE	7AF5
CPC	781A	CUBE	7AB6
CURSOR	7943	ENTER	779A
FIX	7584	FLOW	7DFB
GINVERT	7F49	GRMOD	7306
GSCREEN	785E	HEADER	7E9A
INVERT	78A1	NOSNOW	7CCF
PSCREEN	7842	SHAPE	7972
SNOW	7CF4	SPEED	7EFF
SPLOT	7A3F	TXMOD	7302
UNFLOW	7E4D	VLIST	7505



These computerized flight simulators are just a bit too realistic— especially for crash landings . . .

****
DE POTENTIOMETER deel 2 door j.Bronzwaer
****

Hierbij dan het beloofde vervolgprogramma over potentiometers , te beginnen met een correctie op het artikel in het Bronsgroen Eikeltje 1985 no.4 pagina 3 t/m 8:

Het verhaal over de listing met FLIST neemt u maar voor lief ,de listing werd niet uitgevoerd met FLIST , als u maar begrepen heeft dat ik met mijn te publiceren programma's liefst binnen het standaard (stapelen = standaard) geheugen blijf. Ik ga er daarbij van uit dat nog vele clubleden nog niet over een 16K-kaart beschikken (moet u toch voor gaan sparen !!!).
Voor het hieronder gepresenteerde programma geldt dus weer:
Zorg dat u het krijgt op een clubavond via de gebruikelijke kanalen of type het zelf in zonder verspilling van geheugen (uiteraard alleen indien het programma u interesseert.)

"Potentiometer deel 2" is weer in gewone atombasic geschreven maar vereist de aanwezigheid van JOSBOX of AXR-1 . Het programma gaat dit keer echt tot #4000 dus let op met intypen. Eventueel kunt U de dimpointer naar #9800 zetten (?#23=#00;?#24=#98), maar dat is dus eigenlijk niet nodig.
Het programma berekent en tekent de stroomverdeling en vermogenverdeling binnen de potentiometerschakeling. Interessant mag wel wezen dat er een enorm rekenwerk aan een vrij simpele schakeling wordt verricht. Ik gebruik het programma dan ook als demonstratiemodel. De grafieken hebben echter hun beperkingen : Er worden telkens (en dit gebeurt ten gunste van de nauwkeurigheid nogal traag) drie stromen of drie vermogens getekend waarvan telkens de belangrijkste (totaalwaarde) de maximale verticale uitslag van de kromme bepaalt . Extreme belastingen zullen dan ook niet fraai worden uitgetekend . Ik meen me hiervoor te kunnen verontschuldigen met het feit dat een grafiek een beeld moet weergeven , getallen halen we immers uit tabellen . Aan de tabellen is dan ook de nodige zorg besteed.
Het programma laat voor genoemde schakeling duidelijk aan de hand van getallen en grafieken zien waarom potentiometers meestal overbelast worden bij hoge loperstanden en tevens laat het zien dat overbelasting al kan optreden terwijl het maximaal vermogen nog lang niet is bereikt.
Het programma is grotendeels hetzelfde gestructureerd als het eerder geplubiceerde eerste deel.
Het lijkt mij dan ook niet zinvol alle routines en labels te behandelen , alleen nieuws dus.

INITIALISATIE :

Regel 90 en 100 poken een stuk assembler (objectcode) in het programma waarmee vanaf de derde regel het scherm wordt schoongeveegd. U komt dit bij de tabellen wel tegen.

GEGEVENS + GRENSWAARDEN (label b) :

Het programma produceert hier de ingevoerde gegevens , de maximaal toelaatbare stroom , de minimaal toegestane belastingsweerstand en zijn conclusie omtrent eventuele overbelasting . Dit laatste komt in de grafiek terug .

TABEL DEELSTROMEN (label c) :

Het programma berekent de drie stromen in de potentiometer voor een door uzelf te kiezen aantal stappen :
I(T) = totaalstroom = stroom door het gedeelte boven de loper
I(B) = stroom door de belastingsweerstand Rb
I(P) = stroom door het gedeelte onder de loper
Volgens Kirchhoff 1 geldt : I(T) = I(B) + I(P) .

GRAFIEK DEELSTROMEN (label d) :

De drie stromen in het circuit worden uitgetekend in een vast assenstelsel met schaalverdeling in procenten : horizontaal de loperstand in % , verticaal de stroom in % van de totaalstroom I(T) . Treedt er overbelasting op bij een bepaalde loperstand , dan wordt dit in de grafiek gesignaleerd.

TABEL DEELVERMOGENS (label e) :

Het programma berekent de drie deelvermogens in de schakeling voor een door uzelf te kiezen aantal stappen . Pas op , hier ligt het even anders :
P(T) = het totaal door de schakeling opgenomen vermogen
P(B) = het door de belastingsweerstand opgenomen vermogen
P(P) = het totaal door de potentiometer gedisspeerde vermogen t.w. het vermogen boven de loper (dit kan sterk oplopen met de loperstand) + het vermogen onder de loper (dit kan zelfs bij overbelasting nog zeer klein zijn) .

GRAFIEK DEELVERMOGENS (label f) :

De verticale as is uitgedrukt in % van het totaalvermogen , het hier gehanteerde systeem is dus hetzelfde als bij de deelstromen.

ASSENSTELSEL (label g) :

Deze routine verzorgt het afdrukken van de bijschriften in de grafiek en het achtereenvolgens plotten van de assen , de assenverdeling en de drie karakteristieken .

RASTER (label r) :

Als de grafieken zijn getekend kan , om een iets nauwkeuriger aflezing van de grafieken te verkrijgen , een grafisch raster op het assenstelsel worden getekend . Niet zo fraai helaas , voor exacte getallen zij naar de tabellen verwezen .

HOOFDBEREKENINGEN (label h) :

Deze routine wordt zowel bij de berekeningen voor de tabellen als de berekeningen voor de grafieken doorlopen , ze zorgt helaas voor enige traagheid maar dat is niet te voorkomen mits men concessies doet aan de nauwkeurigheid (en dat ligt mij niet) .

Voor elke loperstand %L wordt berekend :
regel 1320 : de weerstandsverdeling v/d potmeter
regel 1330 : de parallelschakeling van de belastingsweerstand en het potentiometergedeelte onder de loper
regel 1340 : de uitgangsspanning
regel 1350 : de totaalstroom

TABELLEN (label t) :

Een routine die de zaak wat onoverzichtelijk maakt maar wel een geheugenbesparing opleverde . Bovendien voor mij weer leerzaam , er wordt met strings geknoeid . Deze routine verzorgt de juiste kop boven de tabellen.

INTEGER VARIABELEN :

C=plaats van de cursor op de regel (#E0) ; J= vermenigvuldigingsfactor om Ampere en Watt resp. in milliampere en milliwatt om te zetten ; K= hulpvariabele INKEY-statement ; L= loperstand voor FOR NEXT-lussen ; M= gekozen menu-regel ; P= regelteller voor de tabellen ; S= aantal gekozen stappen voor de tabellen ; V= het adres vanaf waar de veegroutine verkeerd gegeven input van het scherm veegt ; W= hulpvariabele voor de afrondings -en veegroutines ; X= hulpvariabele FOR NEXT-lussen en horizontale coordinaat bij het plotten van het assenstelsel ; Y= hulpvariabele voor de inputroutine (Y=0 input fout , Y=1 input goed) en verticale coordinaat bij het plotten van het assenstelsel ; Z= hulpvariabele inputroutine en stapvariabele bij het plotten van de grafieken .

FLOATING POINT VARIABELEN :

%A= af te ronden getal ; %B= belastingsweerstand ; %D= het totaal door de schakeling opgenomen vermogen ; %E= het door de belastingsweerstand opgenomen vermogen ; %F= het door de potentiometer totaal opgenomen vermogen ; %I= de totaal door de schakeling opgenomen stroom ; %J= de door de belastingsweerstand opgenomen stroom ; %K= de potentiometerstroom door het gedeelte onder de loper ; %L= de loperstand ; %M = de totaalstroom bij loperstand 100% ; %N= de maximaal toegestane stroom door de potentiometer ; %O= het weerstandsdeel onder de loper ; %P= het maximaal toegestane vermogen voor de potentiometer ; %R= de weerstand van de potentiometer ; %T= de parallelschakeling van de belastingsweerstand en het weerstandsdeel onder de loper ; %U= de ingangsspanning ; %V= de uitgangsspanning ; %W= de minimaal toegestane belastingsweerstand ; %X= de stapvariabele (loperstand) voor het berekenen van het overbelastingspunt en horizontale coordinaat voor het plotten van de grafieken ; %Y= verticale coordinaat voor het plotten van de grafieken

STRINGS :

$B= "stromen" of "vermogens" voor de kop van de tabellen ; $D= "I" of "P" voor de kop van de tabellen ; $E= "mA" of "mW" of "A" of "W" voor de kop van de tabellen ; $I = inputstring , zie inputroutine

FLOATING POINT ARRAYS :

%YY(1-3)= de verticaal berekende waarden voor de grafieken , zijnde respectievelijk totaalstroom , belastingsstroom en potmeterstroom . Ze worden voor de vermogens op dezelfde wijze gebruikt .

TENSLOTTE :

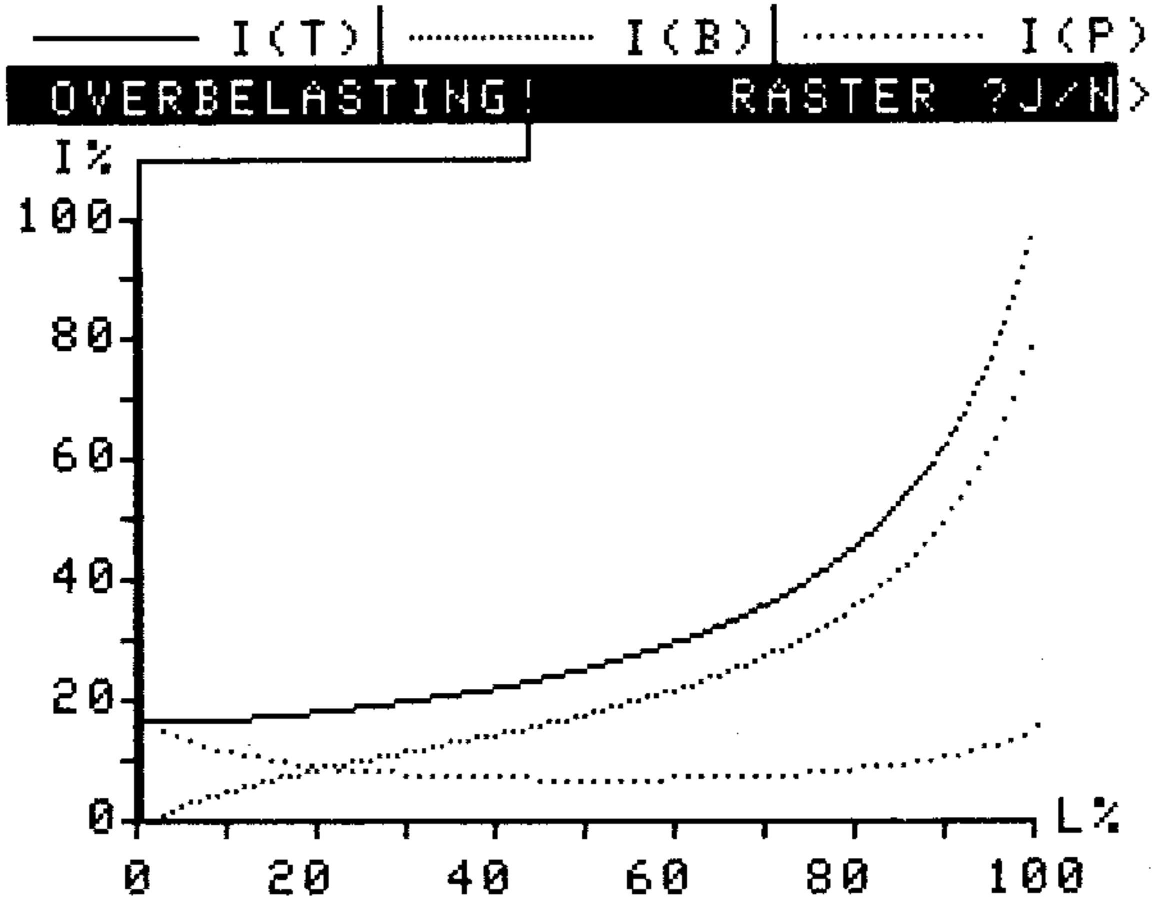
Ik ben mij ervan bewust dat het hoog tijd wordt de P-CHARM utility voor dit soort programma's te gaan gebruiken en ben daar reeds mee bezig. De extra benodigde statements waarmee CHARL DE MOOR mij bijzonder geholpen heeft , kunnen met P=CHARM op een charmante wijze worden gemaakt , ook GRMOD waardoor ik voor eens en altijd de AXR-1 kan inruilen voor een beter gestructureerde manier van programmeren .Ik kan iedere beginner dan ook aanraden dit zo snel mogelijk te gaan doen . Vereist is echter wel dat Uw ATOMgeheugen wordt uitgebreid . Ikzelf heb voor nog geen 100 gulden een AQUARIUS-moduul gemonteerd.
Voor geinteresseerden : veel plezier met dit programma!

```
  10REM POTENTIOMETER (POTM.2)
  20REM ====================
  30REM J.BRONZWAER JUNI 1985
  40REM AXR1 OF JOSBOX NODIG!
  50REM GEHEUGEN: #2900-#3FFF
  60REM initialisatie
  70PL.68;C=#E0;?#B002=#FE;O.G.70
  80FDIM%YY(3);DIMB9,D1,E5,I10,O8;$O="ONEINDIG"
  90!#3FF0=#20A900A2;!#3FF4=#9D80809D
 100!#3FF8=#D0E88100;!#3FFC=#FFFF60F7
 110REM inleiding
 120CLEAR0;MOVE9,46;DRAW54,46;DRAW54,40;DRAW9,40;DRAW9,46
 130P.$30';?C=6;P."DE POTENTIOMETER (2)"'''
 140P."DIT PROGRAMMA BEREKENT DE STROOM"
 150P."-EN VERMOGENVERDELING BINNEN EEN"
 160P."BELASTE  POTENTIOMETERSCHAKELING"
 170P."BIJ VERSCHILLENDE  LOPERSTANDEN."'
 180P."DE RESULTATEN WORDEN WEERGEGEVEN"
 190P."IN TABELLEN EN GRAFIEKEN."''
 200P."BEKIJKT U OOK EENS HET PROGRAMMA"
 210P."""" DE POTENTIOMETER (1) """" " DAT VAN"
 220P."DEZE SCHAKELING DE REGELKARAKTE-"
 230P."RISTIEK BEPAALT.";MOVE40,1;DRAW61,1
 240?C=29;P.">>";LI.#FFE3
 250REM schema
 260CLEAR0;P.$30';?C=17;P."100%"';?C=5;P."+"';?C=2
 270P."IN";?C=15;P."P";?C=17;P.$95';?C=20;P."L"'
 280?C=26;P."R";?C=29;P."UIT";?C=17;P."0%"''
```

```
290MOVE13,25;DRAW13,35;MOVE12,35;DRAW14,35;MOVE10,37
300DRAW16,37;MOVE13,37;DRAW13,46;DRAW31,46;DRAW31,42
310MOVE29,42;DRAW33,42;DRAW33,29;DRAW29,29;DRAW29,42
320MOVE31,29;DRAW31,25;MOVE13,25;DRAW53,25;DRAW53,28
330MOVE51,28;DRAW55,28;DRAW55,34;DRAW51,34;DRAW51,28
340MOVE53,34;DRAW53,37;DRAW35,37
350REM invoer
360P."GEGEVENS :"'';MOVE0,17;DRAW15,17
370P."POTMETERWEERSTAND   ";V=#8175;GOS.i;IFY=0;G.370
380%R=ABSVAL$I;R=%R;FIF%R<1E-2;GOS.v;G.370
390P."POTMETERVERMOGEN    ";V=#8195;GOS.i;IFY=0;G.390
400%P=ABSVAL$I;FIF%P<1E-2;GOS.v;G.390
410P."BEL.WEERST.  R(B)   ";V=#81B5;GOS.i;IFY=0;G.410
420%B=ABSVAL$I;FIF%B<1E-2;GOS.v;G.410
430P."INGANGSSPANNING     ";V=#81D5;GOS.i;IFY=0;G.430
440%U=ABSVAL$I;M=1;FIF%U<1E-3;GOS.v;G.430
450%N=SQR(%P/%R)
460mREM menu
470TXMOD;CLEAR0;P.$30;@=4
480P."       M E N U"''"       ======="''
490P." 1.  GEGEVENS EN GRENSWAARDEN."''
500P." 2.  BEREKENING DEELSTROMEN."''
510P." 3.  GRAFIEK DEELSTROMEN."''
520P." 4.  BEREKENING DEELVERMOGENS."''
530P." 5.  GRAFIEK DEELVERMOGENS."''
540P." 6.  OPNIEUW BEGINNEN."'';?C=5
550P.$128"uw"$128"keuze"$159$128
560DOKEYK;U.K<>0
570IFK=54ORM=0;RUN
580IFK=49;M=1;G.b
590IFK=50;M=2;G.c
600IFK=51;M=3;P=1;G.d
610IFK=52;M=4;G.e
620IFK=53;M=5;P=1;G.f
630P.$7;G.560
640bREM gegevens+grenswaarden
650CLEAR0;P.$30"GEGEVENS:"'';MOVE0,44;DRAW16,44
660FP."R(P)  = "%R;?C=27;P."OHM"'
670FP."P(P)  = "%P;?C=26;P."WATT"'
680FIF%B=2E9;P."R(B)  =    "$0;?C=27;P."OHM"';G.700
690FP."R(B)  = "%B;?C=27;P."OHM"'
700FP."U(IN) = "%U;?C=26;P."VOLT"''
710MOVE 0,28;DRAW59,28;P."DE MAXIMAAL "
720P."TOELAATBARE STROOM"'"DOOR DE POTMETER : "
730%A=%N*1E3;GOS.a;?C=28;P."MA"''
740P."VOOR DE HIER GELDENDE GEGEVENS"'
750P."BEDRAAGT R(B)MIN : "
760FIF%N<=%U/%R;?C=22;P.$0'';G.780
770%W=%U/(%N-%U/%R);%A=%W;GOS.a;?C=27;P."OHM"''
780P.$128"momentje"$129$128;?C=0;%X=10;%L=-10
790DO%L=%L+%X;GOS.h;FIF%I>=%N;%L=%L-%X;%X=%X/10;GOS.h
800FIF%L>100;P."ER TREEDT GEEN OVERBELASTING OP!";G.830
810FU.(%N-%I)<(%I/1E4);P."OVERBELASTING ZAL OPTREDEN BIJ"'
820P."EEN LOPERSTAND VAN:";%A=%L;GOS.a;?C=29;P."%"
830P.$7$7;LI.#FFE3;G.m
840cREM tabel deelstromen
850GOS.t;P=-1;%L=0;F.L=0TO100S.100/S;P=P+1
860IFP=11;LI.#FFE3;P=0;LI.#3FF0;P.$30''''
```

```
 870%L=L;GOS.h;%I=J*%I;%J=J*%V/%B;%K=%I-%J;P.%L
 880?C=6;%A=%I;GOS.a;?C=15;%A=%J;GOS.a;?C=24;%A=%K;GOS.a
 890P.';N.;LI.#FFE3;G.m
 900dREM grafiek deelstromen
 910GOS.g;%L=100;GOS.h;%M=%I;F.Z=1TO3;%L=-Z*0.5;DO%L=%L+Z*0.5
 920GOS.h;%YY(1)=%I;%YY(2)=%V/%B;%YY(3)=%I-%V/%B
 930%X=2*(%L+0.5*Z)+28;%Y=%YY(Z)*120/%M+28;PLOT13,%X,%Y
 940IFZ=1;IFP=1;GOS.o
 950FU.%X>=228;N.;GOS.r;LI.#FFE3;G.m
 960eREM tabel deelvermogens
 970GOS.t;P=-1;%L=0;F.L=0TO100S.100/S;P=P+1
 980IFP=11;LI.#FFE3;P=0;LI.#3FF0;P.$30''''
 990%L=L;GOS.h;%D=J*%U*%I;%E=J*(%V*%V)/%B;%F=%D-%E;P.%L
1000?C=6;%A=%D;GOS.a;?C=15;%A=%E;GOS.a;?C=24;%A=%F;GOS.a
1010P.';N.;LI.#FFE3;G.m
1020fREM grafiek deelvermogens
1030GOS.g;%L=100;GOS.h;%M=%U*%I;F.Z=1TO3;%L=-0.5*Z
1040DO%L=%L+0.5*Z;GOS.h
1050%YY(1)=%U*%I;%YY(2)=%V*%V/%B;%YY(3)=%YY(1)-%YY(2)
1060%X=2*(%L+0.5*Z)+28;%Y=%YY(Z)*120/%M+28;PLOT13,%X,%Y
1070IFZ=1;IFP=1;GOS.o
1080FU.%X>=228;N.;GOS.r;LI.#FFE3;G.m
1090gREM assenstelsel
1100GR.;C?1=0;P." ";$E=" (mA)";$D="I";IFM=5;$E=" (mW)";$D="P"
1110?C=6;P.$D"(T)";?C=17;P.$D"(B)";?C=28;P.$D"(P)"'
1120?C=1;P.$D"%"'"100"''" 80"''" 60"''" 40"''" 20"''"  0"
1130?C=29;P."L%"'"    0    20    40    60    80    100"$30'
1140DOP.$128;U.C.=32;MOVE83,179;DRAW83,192;P.$11
1150MOVE171,178;DRAW171,192;MOVE6,185;DRAW42,185
1160X=90;DOPLOT13,X,185;X=X+2;U.X>=131
1170X=178;DOPLOT13,X,185;X=X+3;U.X>=218
1180PLOT4,28,28;PLOT1,200,0
1190X=28;F.Y=0TO10;PLOT4,X,28;PLOT1,0,-3;X=X+20;N.
1200PLOT4,28,28;PLOT1,0,120
1210Y=28;F.X=0TO10;PLOT4,28,Y;PLOT1,-3,0;Y=Y+12;N.;R.
1220rREM raster
1230?C=20;P."raster"$128$159"j"$143"n"$7$7
1240DOKEYK;U.K<>0;IFK<>74;G.m
1250X=28;F.Y=0TO10;PLOT4,X,148;PLOT1,0,-120;X=X+20;N.
1260Y=28;F.X=0TO10;PLOT4,228,Y;PLOT1,-200,0;Y=Y+12;N.;R.
1270oREM overbelasting
1280FIF%I<%N;R.
1290P=0;?C=1;P."overbelasting"$129$7$7
1300MOVE%X,28;DRAW%X,160;DRAW116,160;DRAW116,168;R.
1310hREM hoofdberekeningen
1320%O=%R*%L/100
1330%T=%O*%B/(%O+%B)
1340%V=%U*%T/(%T+%R-%O)
1350%I=%U/(%T+%R-%O);R.
1360tREM tabellen
1370$E=" (mA)";$D="I";IFM=4;$E=" (mW)";$D="P"
1380J=1000;$B="STROMEN";IFM=4;$B="VERMOGENS"
1390FIF%U/(%R*%B/(%R+%B))>=10;IFM=2;$E=" (A)";J=1
1400FIF%U*%U/(%R*%B/(%R+%B))>=10;IFM=4;$E=" (W)";J=1
1410CLEAR0;P.$30"BEREKENING DEEL"$B$E'';MOVE0,43;DRAW62,43
1420C?1=0;P."HOEVEEL STAPPEN (1-100) ";GOS.i;S=VAL$I
1430IFY=0ORS<1ORS>100;P.$7;G.1410
1440P.$30''" L(%)      "$D"(T)       "$D"(B)       "$D"(P)"'
```

```
1450F.X=0TO31;P."-";N.;R.
1460aREM afronding
1470%@=@;Z=%(100*ABS%A+.51)-ABS%A*100
1480IFZ>99P.%A+SGN%A".00";R.
1490FIF%A<0;%A=0
1500P.%A".";@=0;IFZ<10;P.0
1510P.Z;@=%@;R.
1520iREM input
1530IN.$I;W=0;Y=1;Z=0
1540IFL.I>10;RUN
1550IF$I="^";M=0;G.m
1560IF$I=$0;$I="2E9";R.
1570IF$I="0";$I="1E-5";R.
1580F.X=0TO L.(I)-1
1590IFI?X=45IFW=1;GOS.v;R.
1600IFI?X=45;W=1;N.;R.
1610IFI?X=46;IFZ=1;GOS.v;R.
1620IFI?X=46;Z=1;N.;R.
1630IFI?X<48ORI?X>57;GOS.v;R.
1640N.;R.
1650vREM vegen
1660F.W=0TOL.I;?(V+W)=#20;N.;Y=0;P.$13$11$7;R.
1670END
```



```
************************************************************************
**********                  IRQ-STATEMENT                    **********
************************************************************************
```

Met dit programma is het mogelijk om een subroutine op interruptbasis te laten lopen.
Het statement kan worden opgenomen in de p-charmetabel of in een eigen schakelsoft.
Het programma moet ingetikt zoals het er staat omdat er tijdens het aanroepen iets veranderd wordt in het programma zelf !!!
Het aanroepen gebeurt op de volgende manier :

IRQ #XXXX,A   Met #XXXX een hexadecimaal getal waar de subroutine staat
met A een getal tussen 0 en 255 voor de tijd

Veel plezier met dit statement.

JUVENILE-SOFT (C) 1985
=====================

```
 10 REM IRQ-STATEMENT DOOR ANDY CLAESSEN
 20 DIM LL10;FOR I=0 TO 10;LLI=-1;N.
 30 !#6800=#80C6E3FF;T=#6803;A=#6880
 40 P.$21
 50 P=A;GOS.a;P=A;GOS.a
 60 $T="IRQ";T=T+L.T;?T=LL2/256!#80;T?1=LL2%256;T?2=#80;T=T+2
 70 A=P;T!1=A;?#3FC=#68;P.$6
 80 END
 90aP=A;Q=A
100[
110:LL2
120 JSR#C78B;JSR#C231;JSR#C4E1
130;\ inlezen van de variablen
140 LDX@#00;STX#04
150 LDA#16,X;STAQ+#55;LDA#25,X;STAQ+#56
160;\adres van irqprogramma
170 INX
180 LDA#16;X;STA#21C;STA#21D
190;\ om de hoeveel keer
200 SEI
210;\ ga nu waarden poken
220 LDA@#40;STA#B80B;LDA@#C0;STA#B80E;LDA@#50;STA#B804
230 LDA@#FF;STA#B807;STA#B805
240;\ ga nu irqvector zetten
250 LDA@LL0/256;STA#205
260 LDA@LL0&#FF;STA#204
270 CLI
280 JMP#C55B
290:LL0 LDA#B804;TYA;PHA;TXA;PHA;LDA#21C;CMP@#00;BNE LL1
300 JSR#FFFF;LDA#21D;STA#21C
310:LL1 DEC#21C;PLA;TAX;PLA;TAY;PLA;RTI
320]
330 RETURN
```

```
*****************************************************************
*******              PARAMETERVOORSTELLING               *******
*****************************************************************
```

Deze twee programma's zijn gemaakt voor het vergemakkelijken van de wiskunde. Je hoeft nu alleen maar de functie in te geven vanaf regelnummer 1000 zoals een paar voorbeelden in het programma. D.w.z. een nieuwe functie wordt op een regel gezet met als regelnummer 1000 of 1010 of 1020, etc. Op regelnummer 1005 of 1015 of 1025, etc. moet RETURN komen te staan. (voor een voorbeeld zie het programma zelf).
Bij dit programma worden ook de eventuele asymptoten getekend.
Als men bijvoorbeeld de waarde nul noet mag ingeven dan wordt er een waarde die zeer weinig van nul verschilt ingevuld (zie progamma voorbeeld).
Het programma grafiek 1 : voor normale functies met alleen de y-coordinaat afhankelijk van de x-coordinaat.
Het programma grafiek 2 : voor parameterkromme,dwz zowel de x-coor. als de y-coor. afhankelijk van een variabele in het programma afhankelijk van %T (zie voorbeeld).

Veel plezier met deze (hopelijk) nuttige programma's.

Made by : ANDY CLAESSEN

JUVENILE-SOFT (C) 1985
=====================

```
   10REM GRAFIEKEN
   20REM******grafieken*****
   30REM***MADE BY :      ***
   40REM***ANDY CLAESSEN ***
   50REM***ORANJELAAN 13 ***
   60REM***6074 AT MELICK***
   70REM***04752-4589     ***
   80REM*******************
   90 DIMP-1;P.$21;[JSR#FFE3;STA#80;RTS;];P.$6
  100 %F=0.100001;REM STAFGROOTTE
  110 P.$12;?#E1=0;@=0;P." *** grafiekenprogramma ***"'''
  120 P."MENU :"'
  130 P."WAARDEN PRINTEN     (1)"'
  140 P."GRAFIEK MAKEN       (2)"'
  150 P."ANDERE GRAFIEK      (3)"'
  160 P.'''"(C) 1985 JUVENILE-SOFT"'
  170 K=0
  180 LINK TOP ;A=?#80
  190 IF A=49;G.q
  200 IF A=50;G.w
  210 IF A=51;G.e
  220 GOTO 180
  230iP.$12;?#E1=0;@=0;P." *** grafiekenprogramma ***"'''
  240 P."MENU :"'
  250 FIN."ONDERGRENS      :"%X
  260 FIN."BOVENGRENS      :"%B
  270FIN."VERGROTING     :"%C;%E=%C;%C=%C*10
  280 IN."SCHAAL         :"W
```

```
 290 IN."GRAFIEK NO.    :"D;D=(D-1)*10
 300 IN."OORSPRONG (X)  :"I
 310 IN."OORSPRONG (Y)  :"O
 320 RETURN
 330REM teken de grafiek
 340w
 350 GOS.i
 360 CLEAR4
 370 MOVE0,0;DRAW256,0;MOVEI,0;DRAWI,192
 380 R=150/%C
 390 FOR X=-R TO R
 400 MOVE(I+X*%C),(O-2);DRAW(I+X*%C),(O+2)
 410 MOVE(I-2),(O+X*%C);DRAW(I+2),(O+X*%C)
 420 NEXT
 430 %Z=%X*%C
 440 GOS.(1000+D)
 450 %T=%Y*%C
 460 IF K=0;MOVE(I+(%Z/W)),((%T/W)+O);K=1
 470 DRAW(I+(%Z/W)),((%T/W)+O)
 480 %X=%X+((%F)/%E);%Z=%Z+(%F)*10
 490 FIF %X>=%B;P.$7;LINK#FFE3;RUN
 500 G.440
 510q GOS.i
 520 P.$12;?#E1=0;P."*** getallen ***"''
 530 IN."PRINTER (Y/N)"$#2800
 540 IF $#2800="Y";P.$2
 550 P."X-COORDINAAT :      Y-COORDINAAT :"''
 560 GOS.(1000+D)
 570 FP.%X;P."       ";FP.%Y;P.'
 580 %X=%X+((%F)/%E);FIF %X>=%B;P.$3$7;LINK#FFE3;RUN
 590 G.560
 600e GOS.i;?#B000=#F0;FORX=#8000 TO #81FF S.4;!X=0;N.
 610 G.370
 620REM  functies
1000 %Y=SIN(%X)
1005 R.
1010 %Y=COS(%X)
1015 R.
1020 %Y=TAN(%X)
1025 R.
```

```
  10REM PARAMETERVOORSTELLINGEN VAN KROMMEN
  20REM******grafieken*****
  30REM***MADE BY :      ***
  40REM***ANDY CLAESSEN ***
  50REM***ORANJELAAN 13 ***
  60REM***6074 AT MELICK***
  70REM***04752-4589    ***
  80REM*********************
  90 DIMP-1;P.$21;[JSR#FFE3;STA#80;RTS;];P.$6
 100 %F=0.10001;REM STAPGROOTTE
 110 P.$12;?#E1=0;@=0;P." *** grafiekenprogramma ***"'''
 120 P."MENU :"'
 130 P."WAARDEN PRINTEN     (1)"'
 140 P."GRAFIEK MAKEN       (2)"'
 150 P."ANDERE GRAFIEK      (3)"'
 160 P.'''"(C) 1985 JUVENILE-SOFT"'
```

```
170 K=0
180 LINK TOP ;A=?#80
190 IF A=49;G.q
200 IF A=50;G.w
210 IF A=51;G.e
220 GOTO 180
230iP.$12;?#E1=0;@=0;P." *** grafiekenprogramma ***"''
240 P."MENU :"'
250 FIN."ONDERGRENS      :"%T
260 FIN."BOVENGRENS      :"%B
270FIN."VERGROTING      :"%C;%E=%C;%C=%C*10
280 IN."SCHAAL          :"W
290 IN."GRAFIEK NO.     :"D;D=(D-1)*10
300 IN."OORSPRONG (X)   :"I
310 IN."OORSPRONG (Y)   :"O
320 RETURN
330REM teken de grafiek
340w
350 GOS.i
360 CLEAR4
370 MOVE0,O;DRAW256,O;MOVEI,0;DRAWI,192
380 R=150/%C+ABS(I-128)/%C
390 FOR X=-R TO R
400 MOVE(I+X*%C),(O-2);DRAW(I+X*%C),(O+2)
410 MOVE(I-2),(O+X*%C);DRAW(I+2),(O+X*%C)
420 NEXT
430 GOS.(1000+D)
440 %G=%Y*%C;%Z=%X*%C
450 IF K=0;MOVE(I+(%Z/W)),((%G/W)+O);K=1
460 DRAW(I+(%Z/W)),((%G/W)+O)
470 %T=%T+((%F)/%E)
480 FIF %T>=%B;P.$7;LINK#FFE3;RUN
490 G.430
500q GOS.i
510 P.$12;?#E1=0;P."*** getallen ***"''
520 IN."PRINTER (Y/N)"$#2800
530 IF $#2800="Y";P.$2
540 P."X-COORDINAAT :       Y-COORDINAAT :"''
550 GOS.(1000+D)
560 FP.%X;P."         ";FP.%Y;P.'
570 %T=%T+((%F)/%E);FIF %T>=%B;P.$3$7;LINK#FFE3;RUN
580 G.550
590e GOS.i;?#B000=#F0;FORX=#8000 TO #81FF S.4;!X=0;N.
600 G.370
610REM  functies
1000 %Y=%T-4/%T
1002 %X=%T-1
1005 R.
1010 %Y=1/%T+2
1012 %X=1/((ABS%T^2)-1)
1015 R.
1020 %Y=SIN(%T)*%T
1022 %X=COS(%T)*%T
1025 R.
1030 ETC.
```

Het statement DIAL en de daarbij behorende kies automaat

Om met het laatste tebeginnen, Na een aantal ontwerpjes voor een kiezer in atomnieuws gezien te hebben, ben ik de kast maar eens ingedoken en heb m'n eigen ontwerp te voorschijn gehaald.
Dit ontwerp voldoet volgens mij nog het beste aan de eisen van de PTT,het kies gedeelte zelf is nl. een exacte kopie van een telefoontoestel.
Verder is er ook nog gedacht aan kiestoon detectie.

Het schema is grofweg in drie gedeeltes te verdelen nl.

1.het gedeelte boven de 5volt lijn.Dit is het telefoongedeelte van links naar rechts gezien komt men de volgende onderdelen tegen.
Relais A.;dit is het haakcontact.Op de (p)contacten wordt de telefoonlijn aangesloten,op de (nc) contacten wordt uw eigen telefoon aangesloten,op de (no) contacten komt de kiesunit.
Relais B.dit is het impulscontact en is normaal gesloten, tijdens het kiezen van een telefoon nummer gaat dit afwisselend open en dicht.parallel over dit contact staat een serieschakeling van C1 en R1,dit is een vonkenblus keten.
Relais C.;dit is het kortsluitcontact,dat er voor zorgt dat tijdens de kies impulsen de trafo 1 kortgesloten is; dit is nodig omdat tijdens het kiezen de telefooncentrale een kortsluiting of een open lijn moet zien.
Trafo 1.;deze vervangt de spreek/hoorspoel uit de telefoon.

Ter verduidelinking volgen nu een viertal tekeningetjes wat de tel.centr. electrisch gesproken op een tel.lijn kan 'zien'.

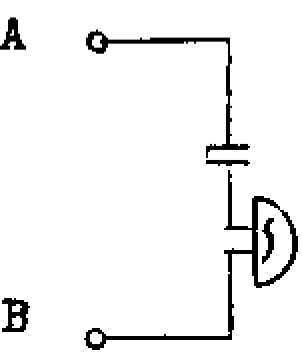


fig.1

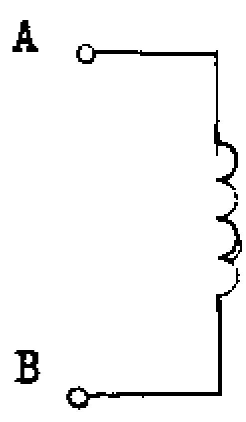


fig.2



fig.3

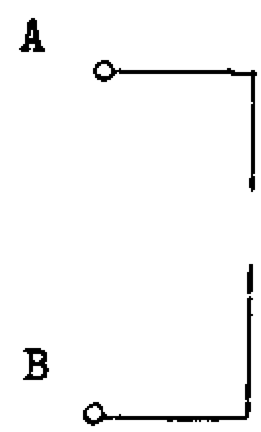


fig.4

figuur 1 is aanwezig als de haak van de telefoon er op ligt, figuur 2 is aanwezig als u aan het spreken bent.
figuur 3 en figuur 4 zijn afwisselend aanwezig als u de kiesschijf teruglaat lopen als de schif weer in rust is is fig.2 weer aan de orde.

2. over het tweede stuk valt weinig te vertellen.
dit is de aansturing van de relais'. Relais 1 wordt aangestuurd via een transistor vanwege z'n stroomverbruik(100mA) relais B. is vanwege de dubbele inverter normaal gesloten, relais C. is normaal geopend.

3. Het laatste gedeelte,dat elektrisch gezien achter TR.1 zit en in het schema rechts,zorgt voor de kiestoon detectie op de lijn.

Het kiesmodem wordt aangelòten op de VIA 6522 port B pen 4-7 dit kan men natuurlijk wijzigen maar dan zal ook de software aangepast moeten worden.

Het statement "DIAL".

Met DIAL is van uit een programma of direct vanaf het toetsenbord het statement aan te roepen.
Het tebellen telefoonnummer moet als een string worden ingevoerd.
Dus : DIAL"085-634842";of DIAL $A;of DIAL $AA(I).

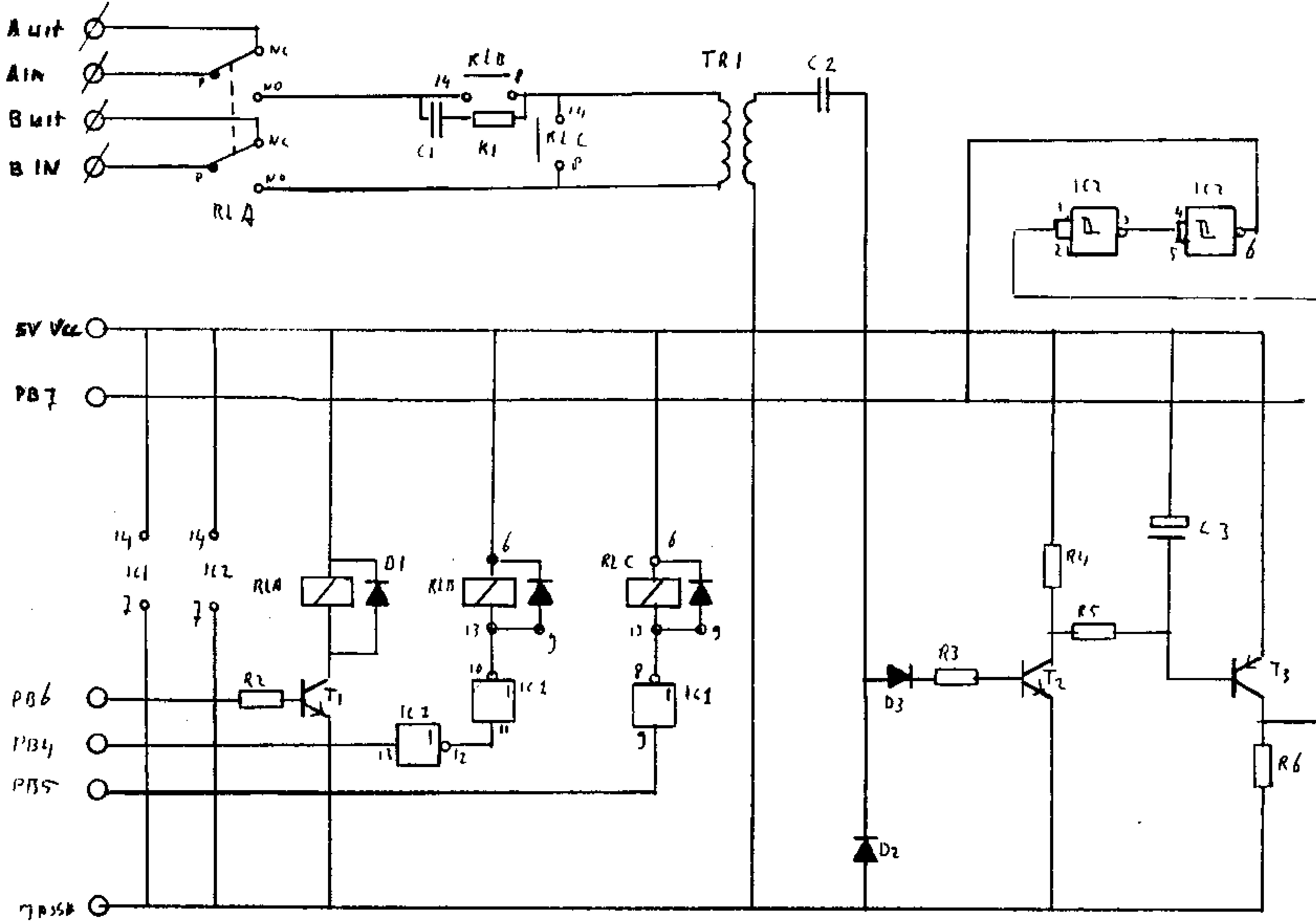
Tijdens het kiezen test het prog. of het karakter een streepje is en/of een getal.
Als het kar. een streepje is wordt er middels een JSR naar de testroutine voor kiestoon gesprongen;als het prog. hier onverhoopt blijft hangen(geen tel.lijn aangesloten) is er middels de control uittekomen.
Als het kar.geen getal is komt het prog. terug met een error melding en wordt het kiezen afgebroken en de tel.lijn weer vrij gemaakt.

Dan volgt nu nog de onderdelenlijst van de kiezer.

R1=560	T1=2N1613
R2=1K3	T2=BC547B
R3=301K	T3=BC557B
R4=1M	
R5=100K	D1=1N4001 oid
R6=1K	D2=1N4001 oid
	D3=1N4001 oid
C1=1uF	
C2=10nF	IC1=7406
C3=4uF,16V	IC2=74132

RL A=printrelais 5 volt 100mA 2xwissel
RL B= clare prme 15002
RL C=

TR1= lijntrafo 1:1 300 Ohm


A uit
A in
B uit
B IN
RL A
RL B
RL C
C1
R1
TR1
C2
IC2
5V Vcc
PB7
IC1
IC2
D1
RLA
RLB
RL C
C3
R4
R5
R2
T1
R3
T2
T3
PB6
PB4
PB5
D3
R6
D2
massa

```
10 PROGRAM DIAL
20P.$15
30 PASS2;GOS.a
40 PASS1;GOS.a
50P.$14
60 END
70aASM-BEGIN OF DIAL ASSEMBLEY
80:BASE            =#400
85:PORT'B=#B001
90:BASIC           =#C55B
100:CR             =#0D
110:STRINGBUFFER   =#140
120:ddrb           =#B802
130:drb            =#B800
140:YSAV           =#8F
150:OSWRCH         =#FFF4
160:WACHT          =#FB83
170:PR'TEXT        =#F701
180.CODE#400
190.OPTION#30
200
210.HARD
220 NOP;NOP;NOP
230 LDA drb\
240 AND@#BF\ ZET HAAKKONTACT
250 STA drb\ AF
260 JMP BASIC
270:START
280 JSR#CEFA
290 STY#34
300 STY#43
310 LDA#53
320 CMP@1
330 BNE KOPY
340 LDA#52
350 CMP@#40
360 BEQ KIES
370:KOPY
380 DEY
390:LOOP1
400 INY
410 LDA(#52),Y
420 STA STRINGBUFFER,Y
430 CMP@CR
440 BNELOOP1
450:KIES
460 LDA ddrb
470 ORA@#70
480 STA ddrb
490 LDA drb
500 ORA@#40
510 STA drb
520 JSR WACHT'KIESTOON
530 LDY@#FF
540 STY YSAV
550:LOOP2
560 INC YSAV
570 LDY YSAV
580 LDA STRINGBUFFER,Y
590 CMP@#2D\VERG.STREEPJE
600 BEQ BEL'OP
610 CMP@CR
620 BEQ EXIT
630 CMP@#30\GEEN'GETAL
640 BCC error
650 CMP@#40\GEEN'GETAL
660 BCS error
670:BEL'OP
680 CLC
690 ADC@#60    \PRINT ACCU
700 JSR OSWRCH\GEINVENTEERD
710 CMP@#8D\VERG.INVERS STREEPJE
720 BNE GO'ON\ZO NIET GA DOOR
730 JSR WACHT'KIESTOON\WEL STREEPJ
        WACHT OP KIESTOON
740 JMP LOOP2\HAAL VOLGENDE GETAL
750:GO'ON
760 SBC@#90\HAAL ER#90VAN AF
770 BNE GEEN'NUL\INDIEN NUL
780 CLC\ TEL ER TIEN BY OP
790 ADC@#A\WANT NUL*TIEN BY DE ptt
800:GEEN'NUL
810 TAY\ZET GETAL IN Y-REG. ALS LOOP TELLER
820 LDA drb
830 ORA@#20\SLUIT KORTSLUITKONTACT
840 STA drb
850 JSR DELAY+6
860:LOOP3
870 ORA@#10\OPEN IMPULSCONTACT
880 STA drb
890 JSR DELAY
900 AND@#EF\SLUIT IMPULSCONTACT
910 STA drb
920 JSR DELAY+3
930 DEY
940 BNE LOOP3\CYFER NOG NIET KLAAR
950 AND@#DF\OPEN KORTSLUIT CONTACT
960 STA drb
970 JMP LOOP2\HAAL VOLGENDE CIJFER
980:EXIT
990 JSR#CD54
1000 RTS
1010:error
1020 JSR PR'TEXT
1030.BYTE#07,#07,#0D,#0A
1040.ASCII'ALLEEN GETALLEN'
1050.BYTE#0D,#0A
1060 NOP
1070 JMP EXIT
1080:DELAY
1090 LDX@#4
1100 BIT#04A2
1110 BIT#1EA2
1120 JMP WACHT
1130:WACHT'KIESTOON
1131 BIT PORT'B
1132 BVS TEST'KIESTOON
1133 PLA
1134 PLA
1135 JMP EXIT
1136:TEST'KIESTOON
1140 BIT drb
1150 BPL WACHT'KIESTOON
1160 RTS
1170.IF (?#28FF&1);.CODE
     BASE;JSR START
1180.END OF SEMBLY
1190 RETURN
```

Voor hen die in het bezit zijn van ram op #7000 of op #A000 of een schakelkaart kan het volgende toch van belang zijn.

Een inleidend verhaaltje
Van de bestaande schakelsystemen zijn het cx-systeem en de branquart schakelsoft de meest geavanceerde. Ze zijn uitgebreid beschreven in Acorn Nieuws 4-4. Zelf werk ik onder branquart.
De schakelsoft staat van #1000 tot #153A onder write-protect en battery-backup, met als sckakelbyte #BFFF en schaduwbyte #9FFF.
Aangezien de DISK-rom #9FA0-#9FDB, #9FE0-#9FE5 en #9FF0-#9FF2 als kladblok gebruikt is voor branquart als werkgebied #9EDD-#9F9F gereserveert. Dus niet in het 1000-blok, want dat staat onder write-protect. Vanaf #1600-#2000 staat een P-charme basic uitbreiding die mij alle error-codes geeft. (zie P-charme handleiding blz.35)

Het cx-systeem
Het cx-systeem heeft zijn kladblok vanaf #9900, tenminste als er nog geen nieuwe versie is uitgekomen.
Zoals je misschien nog wel weet heb je bij het cx-systeem een box op #A000 nodig, de CX-2.7, en wel in voet 2. Deze box is voor een deel alleen in combinatie met de rest van het cx-systeem te gebruiken. Dit vormde een bezwaar om de box als vaste toolbox in mijn systeem op te nemen.

Zelf een toolbox maken
Ik heb dus zelf maar een toolbox gemaakt. Ik zeg niet geschreven, want geen enkel statement heb ik zelf gemaakt, alleen heb ik hier en daar wel wat gesleuteld.
Om een toolbox te maken heb je sources nodig. Behalve diegenen die waren gepubliceerd, en dus in het clubarchief aanwezig waren, heb ik er velen gemaakt met behulp van 'sourcemaker'. Dit is een nuttig hulpprogramma bij het maken van een toolbox. Je laadt een toolbox op #A000, en nog een keer op bijvoorbeeld #4000, en zoekt in de tabel op waar het beoogde statement naar toe linkt, en disassembleert dit gedeelte (in het 4000-blok dus) om te kijken of er geen gekke sprongen gemaakt worden, en waar het eind is. Nu weet je het begin en eindadres wat je bij 'sourcemaker' moet opgeven, die dan verder het werk voor je doet. In de listing kun je, door naar de labels te kijken, nogmaals contoleren of er niet buiten het opgegeven gebied in het A-blok gesprongen is.
Het statement RELOC uit de josbox heb ik overigens niet kunnen gebruiken want dat werkt niet volledig, een JMP wordt niet aangepast aan het nieuwe adresgebied.
De interpretor voor je toolbox leen je gewoon uit een andere box, waarbij je op moet letten waar de tabel begint. Ik ben van mening dat de tabel het beste direct na de interpretor geplaatst kan worden. Het eerste statement assembleer ik tot #B000. Het volgende statement moet dus eindigen op het byte voor het begin van het eerste statement. Met behulp van write-protect en PRINT&P probeer je dat eerst uit. En zo ga je steeds verder.
Met behulp van 'edit', ook zo'n handig programma kun je de tabel vullen nadat je je toolbox eerst op #4000 hebt geladen. Op deze manier maak je het beste gebruik van de ruimte in je toolbox, en brei je de tabel en de statements naar elkaar toe.

Het resultaat
De nieuw gevormde ACE-2.2, de Atom Commands Extension rom versie 2.2, vormt de tegenhanger van de CX-2.7, maar kan in tegenstelling tot de CX-2.7 volledig zelfstandig gebruikt worden.
Uit de CX-2.7 zijn LCASE en USKEY weggelaten, omdat ik daar zo gauw geen source van kon bemachtigen. Tevens zijn de statements FIND, CLS, FILL en PAGE weggelaten omdat deze respectievelijk al aanwezig zijn in p-charme, gags en branquart. Toegevoegt is SHAPE uit de josbox, zodat nu vrijwel alle programma's geschreven voor deze box ge- draait kunnen worden. Tevens het statement DISAS, dat een disas- sembling gecombineert met een ASCII en HEX-dump geeft, verder SEARCH dat de plaats van een opgegeven string of instructie in het geheugen opzoekt en : waarmee je geheugenplaatsen op eenvoudige manier met hexadecimale getallen of strings kunt vullen. HVAR drukt de waarden van de variabelen hexadecimaal af. ACE linkt naar #C558, en kan gebruikt worden om de box voor te schakelen.

Met de komst van deze box lijkt het me, dat het nog aantrekkelijker geworden is om over te gaan op branquart schakelsoft. Dit systeem is sneller en flexibeler.
Een advies dus ook voor diegenen die willen gaan schakelen.
Diegenen die geen schakelkaart hebben, maar wel ram vanaf #7000 - #8000, kunnen gebruik maken van de p-charme uitbreiding die dezelfde statements als ACE-2.2 kent. De tabel staat hier op #7000, terwijl het commando VLIST vanaf #6F00 is bijgevoegd.
Misschien is dit aanleiding voor diegenen die alleen nog maar onder de josbox werken om over te gaan op de p-charme.
Een waarschuwing is nog op zijn plaats. Sommige programma's, met name die van Ronald Boers, linken in de josbox, zodat ze niet direct zonder de josbox gebruikt kunnen worden. Het programma BACKUP heb ik dan ook al omgeschreven voor gebruik onder p-charme. Misschien kan Ronald Boers zelf zijn overige programma's aanpassen voor gebruik onder pcharme, disk of ace.
Verder wil ik iedereen bedanken die al of niet bewust meegewerkt heeft aan de totstandkoming van deze toolkit.
Voor reakties, opmerkingen en vragen kun je terecht bij:

Gerard van den Brandhof
Julianalaan 151
2628 BG Delft
tel.: 015 - 565408

```
ACE-2.2       :Toolbox op #A000
ACE-70        :P-charme uitbreiding met tabel op #7000 (?#3FC=#70)
HELP          :HELPx, geeft indien opgegeven errornummer niet
               bestaat de beschikbare statements uit ACE, wordt
               geplaatst na branquart-schakelsoft van #1560 - #1FFF
BACKUP        )
WELCOME       ):Backup van disk naar tape voor p-charme
EXPLAN.       )
```

Atom Commands Extension rom v-2.2

Checksum van #A000-#AFFF : #97BE
Interpretor: #A000-#A02B
Tabel : #A02C-#A0D1
Statements:

GRMOD	TXMOD	PLAY
SHAPE	FAST	SLOW
DISAS	SEARCH	:
CLC	HVAR	FVAR
FZERO	SCREEN	GSCREEN
FIX	CON	COFF
LIB	CURSOR	HEADER
FLASH	DUMP	EXEC
PIP	ACE	

Gebruik van statements met variabele(n):

```
DIM A5;$A="FP.PI";EXEC A
FLASH 511,1,1
HEADER 6
CURSOR 30,15
FIX\1,8\PI
SCREEN 1
GSCREEN 9
:70CA "ACC-DELFT" of 4C 58 C5
SEARCH "ATOM" of 0D FF of [JMP#C558]
DISAS#A000,
SHAPE#3A00
PLAY A8,B'4,C"2,R1,D#2,E'#4,F"#8,G1
```

Overige statements zonder variabele(n) gebruiken.

GAGS 2.3

Na alle problemen met GAGS 2.1 tijdens het verspreiden, kwam al zeer rap versie V2.2. Deze heeft ook geen lang leven mogen hebben want Yvo Tuk (Rotterdam) kwam al zeer snel met een fout in het SCROLLen in kleurenmode. Terecht, foutje!
Na het verspreiden van het (prachtige) spel POGO, belde Dick Protzman (Deventer) op en meldde dat POGO goed werkte met GAGS V2.0E, maar niet met GAGS V2.2. Na enig speurwerk bleek de nieuwe ATHIT routine fout te lopen bij enkele situaties en ja hoor, POGO barstte van die situaties.
Nu zijn in GAGS V2.3 de problemen hopelijk opgelost (voor altijd?).

Gerrit

GDUMP voor de Star SG-10 printer

De programma's "GDUMP.SG-10.I" en "GDUMP.SUPERDR" werden beide geschreven voor de Star SG-10 printer. Deze printer werd een goed jaar geleden geintroduceerd op de Nederlandse markt, als opvolger van de Star Gemini 10X. De Star SG-10 kan echter aanzienlijk meer: hij biedt de mogelijkheid tot NLQ (Near Letter Quality), is in grote mate compatibel met de printer-standaard voor de IBM-PC en kan grafisch zeer fraaie plaatjes produceren, tot een max. dichtheid van 240 puntjes per inch!
Hoewel deze laatste zogenaamde *Bit Image Mode*, aangeduid met de term *Quadruple Density* (viervoudige dichtheid), ook al in de Gemini 10X zat, is de SG-10 verrijkt met een aantal nieuwe plot-modes, waarvan er 1 de mogelijkheid biedt om horizontaal en verticaal in dezelfde dichtheid te printen (72 puntjes per inch), waardoor een verhouding van 1:1 ontstaat. Op deze zogenaamde *Plotter Graphics* mode zal ik later nog terugkomen.
In totaal beschikt de SG-10 over 7 grafische modes, die met behulp van een zogenaamde *Master Graphics Selection* zeer eenvoudig en direct aan te sturen zijn. Dit is dan ook wat het programma "GDUMP.SG-10.I" doet: het definieert een dump-commando waarmee een 'CLEAR 4' scherm van de Atom op 7 verschillende manieren kan worden getekend. De Star printer kan in 2 modes geschakeld worden, te weten de 'Star' mode en de 'IBM' mode, elk met een eigen serie controle-codes. In dit programma is, net als in de Nieuwe Word Pack Rom (zie A.N. 4.7, pp.72-77), gekozen voor de IBM mode, omdat deze beter aansluit bij de meest gangbare printercodes (o.a. Epson en Logitec).
Het programma is, net als "GDUMP.SUPERDR", geschreven in P-Charme en benodigt de mini-assembler. De routine kan zonder meer als statement in elke box worden opgenomen. Verander daartoe eventueel de instructie 'JMP BASIC' (r. 1550) in 'RTS'.

De syntaxis van het statement 'GDUMP' (kies eventueel een andere naam) is als volgt:

'GDUMP [*mode* [,*masker* [,*hor.verpl.* [,*relocatie-adres*]]]]'

Zoals U ziet, zijn alle 4 parameters optioneel.
Ze zullen nu een voor een worden besproken.

* *mode*

Hiermee wordt de plot-mode geselecteerd.
De gekozen waarde dient tussen 0 en 6 te liggen.
De default-waarde is 0 (60 puntjes per inch horizontaal).

* *masker*

Deze parameter bepaalt de wijze waarop de tekening wordt weergegeven op de printer. Normaal dient U hier ofwel 0 ofwel -1 in te vullen. In het eerste geval wordt alles wat wit is op het scherm zwart op de printer en omgekeerd, in het tweede geval maakt de printer een exacte copie van het scherm. Met andere waarden kunnen speciale effecten worden bereikt: zie bv. het dump-programma voor de Epson printer dat Bram Poot schreef: A.N. 4.4, pp.54-57. De default-waarde is 0.

* *horizontale verplaatsing*

Met deze parameter kan het plaatje een aantal posities naar rechts worden verschoven, bv. om het precies in het midden te krijgen: zie weer het programma van Bram Poot.
De default-waarde is ook hier 0, d.w.z. geen verplaatsing.

* *relocatie-adres*

Deze laatste parameter biedt de mogelijkheid om ook een afdruk te kunnen maken van een tekening die, om welke reden dan ook, niet vanaf #8000 maar elders in het geheugen ligt opgeslagen. De waarde van de parameter wordt beschouwd als het 'high byte' van het startadres van de tekening: bij #40 bv. wordt dan het geheugengebied van #4000-#5800 als tekening beschouwd en dienovereenkomstig afgedrukt! De default-waarde is #80.

De routine zal een foutmelding genereren als de printer niet is ingeschakeld of als de opgegeven plot-mode groter is dan 6. De routine kan worden gestopt met de 'ESC'-toets. De aansturing van het 8e printerbit geschiedt volgens de club-standaard: zie bv. A.N. 5.1, p.58).
De machinecode beslaat 228 bytes. Als startadres is hier gekozen voor #6700. Verander dit eventueel in regel 50. De routine benodigt tijdelijk 8 ZP-bytes: hier is dit #70-#77 (regels 170-230), wat eveneens aangepast kan worden.

Het tweede programma, "GDUMP.SUPERDR", vormt een variant op het eerste: het is een speciale versie die een printerdump kan maken van een plaatje getekend in 'CLEAR 5' (512 puntjes horizontaal bij 384 verticaal) met behulp van het prachtige programma 'SUPERDRAW' van Rob van Dort (zie A.N. 3.2, pp.41-46). De tekening staat dan opgeslagen van #2000-#8000, en is dus...24K bytes groot!
Om de tekening zo nauwkeurig mogelijk te kunnen weergeven, maakt deze routine gebruik van de hierboven reeds aangeduide plot-mode 5 van de Star SG-10 printer: *Plotter Graphics*.
Ook de overige parameters, die in het eerste programma variabel waren, liggen nu vast: het masker is -1 (wit blijft wit, zwart blijft zwart), de horizontale verplaatsing naar rechts bedraagt 3 posities, en het startadres van de tekening ligt nu op #2000.
Voor het overige zijn de 2 routines vrijwel identiek, behalve dat de printer nu geen 256 maar 512 puntjes per regel afdrukt!
De routine is wat korter geworden dan de vorige, namelijk 166 bytes. Als startadres is gekozen voor #500 (regel 70). Zet de routine in elk geval onder #1C00 neer, daar staat hij veilig.
Voor zover ik heb kunnen nagaan, komen de gekozen ZP-bytes (regels 180-200) niet in conflict met de werkruimte van 'SUPERDRAW'. Volg voor het 'implanteren' van de routine in het tekenprogramma verder de aanwijzingen uit bovengenoemd artikel.

Tot slot: deze twee dump-programma's zijn geinspireerd door de goede grafische mogelijkheden van de Star SG-10 printer. Het zal echter niet al te moeilijk zijn om ze ook voor andere printers geschikt te maken. Zie hiertoe met name het door Johan Siegers samengestelde overzicht van printer-controle-codes in A.N. 4.2, pp.87-89. Veel plezier ermee!

```
  10 PROGRAM GDUMP.SG-10.IBM
  20
  30 P.$21;FOR I=0 TO 1;[
  40
  50 .BA #6700
  60 .TA #9800
  70
  80:STX        = #02
  90:ETX        = #03
 100:CR         = #0D
 110:ESC        = #1B
 120:SPACE      = #20
 130:STAR       = #2A
 140:ASCII'3    = #33
 150:SEMICOL    = #3B
 160
 170:GR'MODE    = #70
 180:IMASK      = #71
 190:HPOS       = #72
 200:PT         = #73
 210:PAGCOUNT   = #75
 220:MASK       = #76
 230:GBYTE      = #77
 240
 250:PORT'B     = #B001
 260:PORT'CTRL= #B003
 270:PR'STATUS= #B801
 280:COMMA      = #C231
 290:STAT'END = #C4E4
 300:BASIC      = #C55B
 310:PARAM      = #CF3E
 320:NCHAR      = #F291
 330:BITMASK    = #F7C9
 340:PRDRIVE    = #FEFB
 350:PR'DIR     = #FF10
 360
 370:GDUMP
 380 LDA @#00
 390 STA GR'MODE
 400 STA IMASK
 410 STA HPOS
 420 STA PT
 430 LDA @#80
 440 STA PT+1
 450 LDA @#18
 460 STA PAGCOUNT
 470
 480 JSR NCHAR
 490 DEC #03
 500 CMP @SEMICOL
 510 BEQ CLOSE
 520 CMP @CR
 530 BEQ CLOSE
 540
 550 JSR PARAM
 560 CPY @#07
 570 BCS NLINE+1
 580 STY GR'MODE
 590 JSR COMMA
 600 BCS CLOSE
 610 JSR PARAM
 620 STY IMASK
 630 JSR COMMA
 640 BCS CLOSE
 650 JSR PARAM
 660 STY HPOS
 670 JSR COMMA
 680 BCS CLOSE
 690 JSR PARAM
 700 STY PT+1
 710
 720:CLOSE
 730 JSR STAT'END
 740 LDA PR'STATUS
 750 BMI NLINE+1
 760
 770 \PRINTER ON
 780 LDA @STX
 790 JSR PRDRIVE
 800
 810 \SET LINE SPACING
 820 LDX @24
 830 JSR LINE'SPACING
 840
 850:SPACES
 860 LDX HPOS
 870 BEQ GRAPH
 880 LDA @SPACE
 890:SPACEPRINT
 900 JSR PRDRIVE
 910 DEX
 920 BNE SPACEPRINT
 930
 940:GRAPH
 950 LDA @ESC
 960 JSR SENDBBIT
 970 LDA @STAR
 980 JSR SENDBBIT
 990 LDA GR'MODE
1000 JSR SENDBBIT
1010 LDA @(256%256)
1020 JSR SENDBBIT
1030 LDA @(256/256)
1040 JSR SENDBBIT
1050
1060:NLINE
1070 LDY @#00
1080
1090:BUILD
1100 LDA BITMASK,X
1110 STA MASK
1120:LOOP
1130 LDA (PT),Y
1140 AND MASK
1150 CMP @#01
1160 ROL GBYTE
1170 LDA PT
1180 CLC
1190 ADC @#20
```

```
1200 STA PT
1210 BCC LOOP
1220 LDA GBYTE
1230 EOR IMASK
1240 JSR SEND8BIT
1250 INX
1260 CPX @#08
1270 BNE BUILD
1280 LDX @#00
1290 INY
1300 CPY @#20
1310 BNE BUILD
1320 LDA @CR
1330 JSR PRDRIVE
1340
1350 \ESCAPE KEY?
1360 LDA PORT'B
1370 AND @#20
1380 BEQ QUIT
1390
1400 \READY?
1410 INC PT+1
1420 DEC PAGCOUNT
1430 BNE SPACES
1440
1450 \RESET PRINTER TO
1460 \6 LINES PER INCH:
1470:QUIT
1480 LDX @36
1490 JSR LINE'SPACING
1500
1510 \PRINTER OFF
1520 LDA @ETX
1530 JSR PRDRIVE
1540
1550 JMP BASIC
1560
1570:LINE'SPACING
1580 LDA @ESC
1590 JSR SEND8BIT
1600 LDA @ASCII'3
1610 JSR SEND8BIT
1620 TXA
1630
1640:SEND8BIT
1650 PHA
1660 ASL A
1670 LDA @#03
1680 ROL A
1690 STA PORT'CTRL
1700 PLA
1710 JSR PR'DIR
1720 \RESET GRAPHIC BIT
1730 LDA @#06
1740 STA PORT'CTRL
1750 RTS
1760
1770];P.$6;NEXT I
1780
1790 END
```

```
  10 PROGRAM GDUMP.SG-10.IBM
  20
  30 REM FOR 'SUPERDRAW'!
  40
  50 P.$21;FOR I=0 TO 1;[
  60
  70 .BA #0500
  80 .TA #1C00
  90
 100:STX        = #02
 110:ETX        = #03
 120:CR         = #0D
 130:ESC        = #1B
 140:SPACE      = #20
 150:STAR       = #2A
 160:ASCII'3    = #33
 170
 180:PT         = #70
 190:TEMP       = #72
 200:GBYTE      = #74
 210
 220:PORT'B     = #B001
 230:PORT'CTRL= #B003
 240:BITMASK    = #F7C9
 250:PRDRIVE    = #FEFB
 260:PR'DIR     = #FF10
 270
 280
 290 \START AT #2000
 300 LDA @#00
 310 STA PT
 320 LDA @#20
 330 STA PT+1
 340
 350 \PRINTER ON
 360 LDA @STX
 370 JSR PRDRIVE
 380
 390 \SET LINE SPACING
 400 \TO 8/72 INCH:
 410 LDX @24
 420 JSR LINE'SPACING
 430
 440:SPACES
 450 LDX @#03
 460 LDA @SPACE
 470:SPACEPRINT
 480 JSR PRDRIVE
 490 DEX
 500 BNE SPACEPRINT
 510
 520 \TELL PRINTER TO RECEIVE
 530 \512 GRAPHIC BYTES
 540 LDA @ESC
 550 JSR SEND8BIT
 560 LDA @STAR
 570 JSR SEND8BIT
 580 LDA @#05
 590 JSR SEND8BIT
```

```
600 LDA @(512%256)
610 JSR SEND8BIT
620 LDA @(512/256)
630 JSR SEND8BIT
640
650 LDY @#00
660
670:BUILD
680 LDA @#01
690 STA GBYTE
700 LDA PT
710 STA TEMP
720 LDA PT+1
730 STA TEMP+1
740:LOOP
750 LDA (TEMP),Y
760 AND BITMASK,X
770 CMP @#01
780 ROL GBYTE
790 BCS SEND'BYTE
800 LDA TEMP
810 ADC @#40
820 STA TEMP
830 BCC LOOP
840 INC TEMP+1
850 BCS LOOP
860:SEND'BYTE
870 LDA GBYTE
880 EOR @#FF
890 JSR SEND8BIT
900 INX
910 CPX @#08
920 BNE BUILD
930 LDX @#00
940 INY
950 CPY @#40
960 BNE BUILD
970 LDA @CR
980 JSR PRDRIVE
990
1000 \ESCAPE KEY ?
1010 LDA PORT'B
1020 AND @#20
1030 BEQ QUIT
1040
1050 \READY ? (AT #8000)
1060 INC PT+1
1070 INC PT+1
1080 BPL SPACES
1090
1100 \RESET LINE SPACING
1110 \TO 12/72 INCH:
1120:QUIT
1130 LDX @36
1140 JSR LINE'SPACING
1150
1160 \PRINTER OFF
1170 LDA @ETX
1180 JMP PRDRIVE
1190
1200 \SUBROUTINES
1210
1220:LINE'SPACING
1230 LDA @ESC
1240 JSR SEND8BIT
1250 LDA @ASCII'3
1260 JSR SEND8BIT
1270 TXA
1280
1290:SEND8BIT
1300 PHA
1310 ASL A
1320 LDA @#03
1330 ROL A
1340 STA PORT'CTRL
1350 PLA
1360 JSR PR'DIR
1370 \RESET GRAPHIC BIT
1380 LDA @#06
1390 STA PORT'CTRL
1400 RTS
1410
1420];P.$6;NEXT I
1430
1440 END
```

```
10 REM    $$$  vinger   $$$
20 REM    GEMAAKT DOOR:
30 REM    HANS BROEKSMA
40 REM    LEEUWARDEN
50 REM VOOR EEN ANDERE SNELHEID DE WACHTLUSSEN VERLENGEN
60 PRINT $12
70 REM NA ELKE PAGINA TEKST MOET ER EEN GOSUB a  VOLGEN.
80 REM VOORBEELD:
90 FOR A=1 TO 16:P." HIER KOMT EEN TEKST TE STAAN"';N.:GOS.a
100 P.$12;F.A=1 TO 16;P." HIER KOMT EEN ANDERE TEKST"';N.:GOS.a
9999 END
10000aFOR A=#8000 TO #81FF STEP 32:?A=#6A:N.
10010 FOR A=#8000 TO #81FF STEP 32
10020 ?A=#4A:F.T=0TO25:WAIT:N.
10030 ?A=#42:F.T=0TO25:WAIT:N.
10040 ?A=#20:F.T=0TO25:WAIT:N.:N.:RETURN
```

*** CMOS-DIS SUPPLEMENT ***

ALGEMEEN

In de toelichting op blz. 37 van AN 7 1985 staan een paar foutjes:

```
  80 P moet A zijn
 182 B moet D zijn
3460 staat een " teveel
```

Verder laat ook de DISASS-subroutine de opcode in de geheugenplaats byt1 als uitvoer achter.
Bij toepassing van DISASS als losse subroutine kunnen tevens de regels 740, 750 en 760 vervallen.

STATEMENTS: DIS EN CDIS

Het kan handig zijn om naast de CMOS-disassembler ook nog over een normale disassembler te kunnen beschikken. Uitgaande van de oorspronkelijke listing (CMOS-DIS 1.0) is dit met de nevenstaande wijzigingen c.q. aanvullingen eenvoudig te realiseren.
Voor het objectcode-gedeelte zijn nu 499 (#1F3) bytes aan geheugenruimte nodig.

De subroutines XDISASS en XSEARCH hebben nu een extra invoer-parameter in de vorm van de carry-flag:

C=0 --> 6502
C=1 --> R65C02

```
SYNTAX:  DIS expressie
         CDIS expressie
```

VOORBEELD:

```
DIS #XXXX       vanaf #XXXX tot #0000
DIS #YYYYXXXX   vanaf #XXXX tot #YYYY
```

Beide statements zijn te onderbreken met de ESCAPE-toets.

```
  10 PROGRAM DIS/CDIS.STAT
  80 D=A    ;REM BEGIN-ADRES
 181 $T="DIS";T=T+LEN(T)
 182 ?T=DIS/256|#80;T?1=DIS%256;T=T+2
 183 $T="CDIS";T=T+LEN(T)
 184 ?T=CDIS/256|#80;T?1=CDIS%256
 185 T?2=#80;T=T+2
 186 A=P;T!1=A
 220 PRINT "BEGIN "&D'

 590:DIS
 591 JSR getadres
 592 JSR START
 593 JMP #C55B
 594:START
 595 JSR DISASS
 596 BNE START
 597
 598:CDIS
 599 JSR getadres
 600 JSR CSTART
 601 JMP #C55B
 602:CSTART
 603 JSR CDISASS
 604 BNE CSTART
 605
 606:getadres
 607 JSR #C4E1
 608 LDY @ptr
 609 LDX #04
 610 JMP #C3CD
 620
 630:DISASS
 631 CLC
 632 BCC XDISASS
 633
 634:CDISASS
 635 SEC
 636
 637:XDISASS
 640 JSR XSEARCH

1800:XSEARCH

1865 BCC OLDSEARCH

3400 !V=M;LINK CDISASS
```

HET RACK

Doel van het rack:

1 De flexibilteit van de Atom opdrijven voor I/O.

2 De geheugen capaciteit opdrijven tot 256kb.

3 Meerdere program partities om multi tasks te kunnen draaien.
Hiervoor moet het geheugen page reloceerbaar zijn.

Welke kaarten zijn hiervoor gebruikt.

1 Er werd gekozen voor het Electuur bus systeem, de reden hiervoor was dat er reeds verschillende kaarten hiervoor op de markt aanwezig waren die mits een kleine aanpassing bruikbaar waren in het systeem.

2 Interface kaarten werden ontworpen en geschakeld tussen de Atom en het rack,alsook enkele kleine modificaties aangebracht in de Atom.

3 Aan de bovengaande punten werdvoldaan door het M.E.M. bord.
Dit I/o bord laat toe elke memory page van 1kb te reloceren op eender welke plaats in een block van 64kb.
Ook wordt hiermede voorzien in de tegemoetkoming van multi tasks, doordat deze kaart 16 verschillende memory maps toelaat te creeeren en te executen.

4 Er werden twee 64kb dyn. ram board en drie 16 kb dyn. ram boards van Electuur gemaakt.
Verder werden nog vier 64 kb universele geheugenkaarten van Electuur gebouwd, die hier gebruikt worden als 16 kb rom boards.

5 Een disk controller werd gebouwd en de software aangepast aan het systeem.

6 Een Eprommer werd gebouwd (Electuur) om de nodige software in Eproms te kunnen opslaan.

7 Verder werd nog een Parrallel printer real time clock kaart en RS232-C kaart ontworpen.

8 Een 80 colloms kaart van electuur werd aan het systeem toegevoegd.

9 Zijn nog in ontwikkeling een D.M.A. kaart, IEE488 kaart,HPIL kaart en een heel snelle A/D kaart.

10 Software: op dit ogenblik draaien P-charm, Gags, Axr1, Disatom, Wordpack, P-charm libery en Eigen libery in #A000 door elkaar, zonder in elkaars vaarwater te zitten,dit door een goede schakelsoft. Hier over later meer in de volgende Atom nieuws letters.

11 Meerdere I/O roms zijn mogelijk in #E000 zodat een goede komunikatie met de biuten wereld kan worden opgebouwd.

12 Indien mogelijk nog een goede grafische kaart waardoor het volgende punt ook in vervulling kan gaan. Dit word de grafische kaart van electuur.

13 De processor in het rack.(ook electuur).

14 Een goed KEYBORD om dit alles aan te vullen. Ook deze wens zal snel in vervulling gaan.

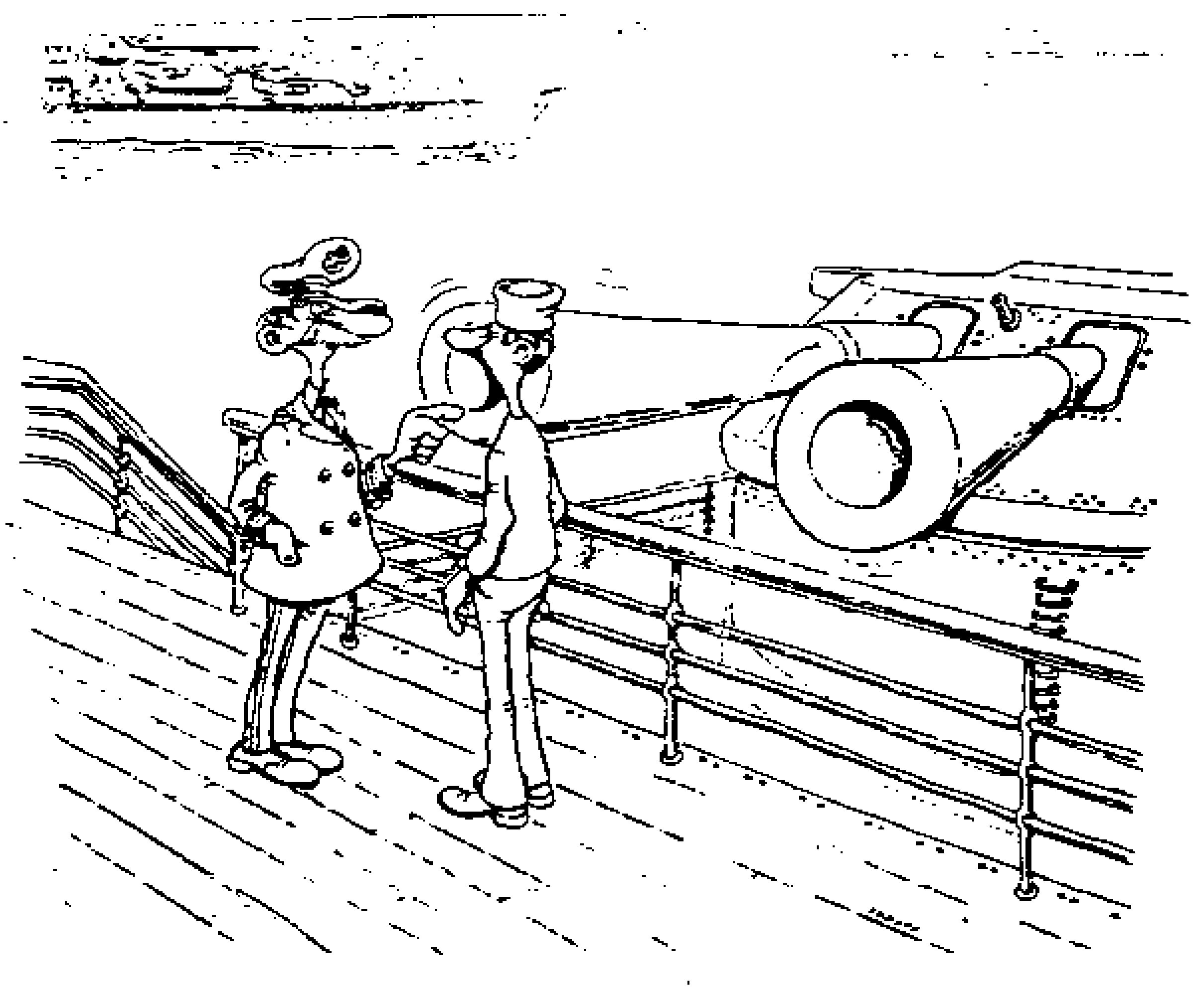
15 Multi task software schrijven.

16 Verder zien we nog wel.

17 De atom moet steeds compatibel blijven met het originele sijsteem, zodat software van anderen ook op dit systeem kunnen draaien. Het is echter steeds een uitdaging om door andere geschreven software die op een voor hun aangepast schakel systeem werkt op dit systeem aan het werken te krijgen.(zoals PCHARM op #1000 etc).
Tot op heden is dit nog steeds gelukt zonder de schakelsoft te hebben moeten herzien.

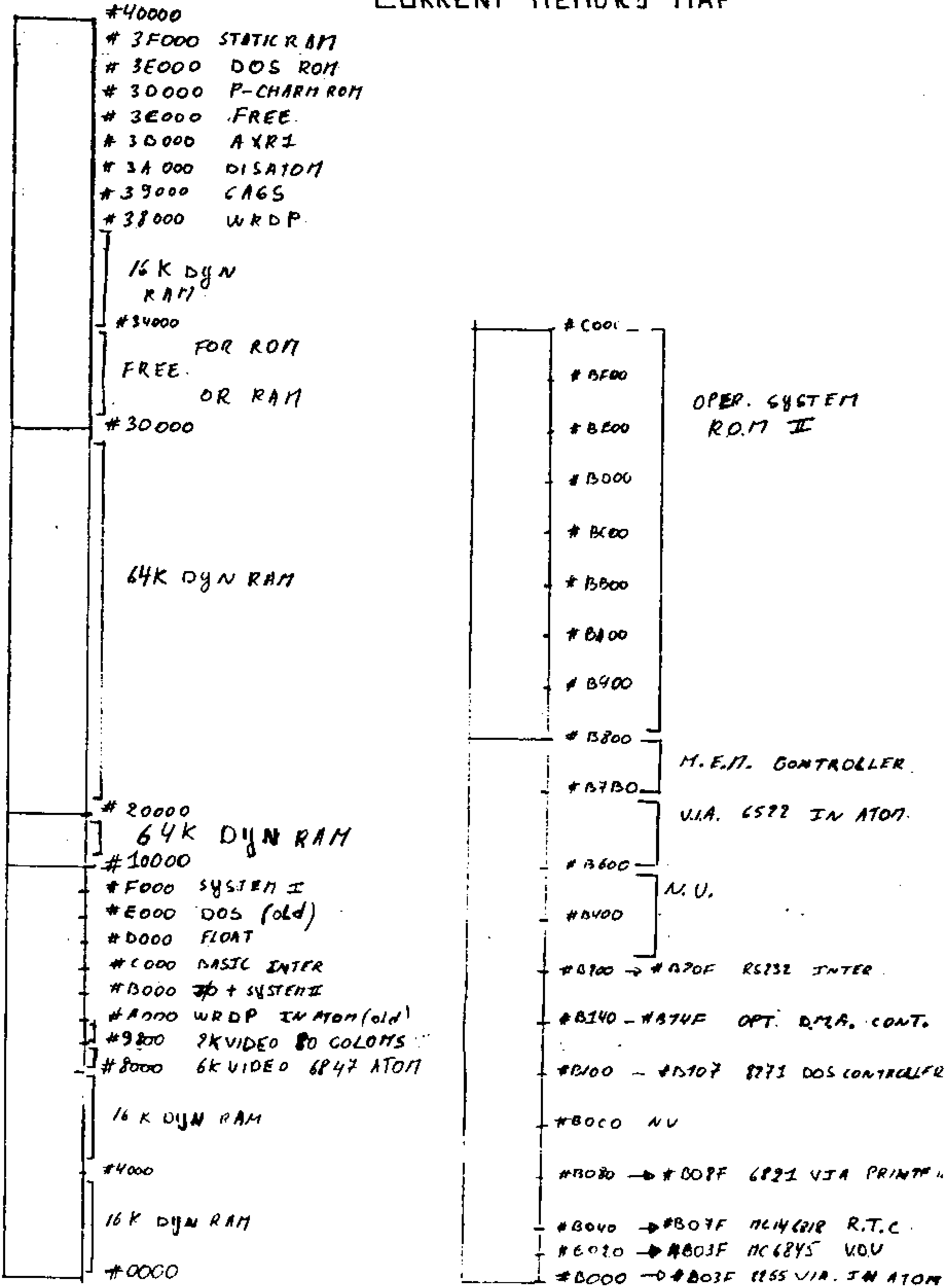
P.S. Toegevoegt vind u nog de huidige memory map van het systeem dat tot op heden met voldoening draaid.
Indien er vragen zijn over schemas of andere dingen kunt u me kontakteren op club vergaderingen, per brief of eventueel telefonisch (adres bekend op de redaktie).

Michel Van leuven A.C.B.

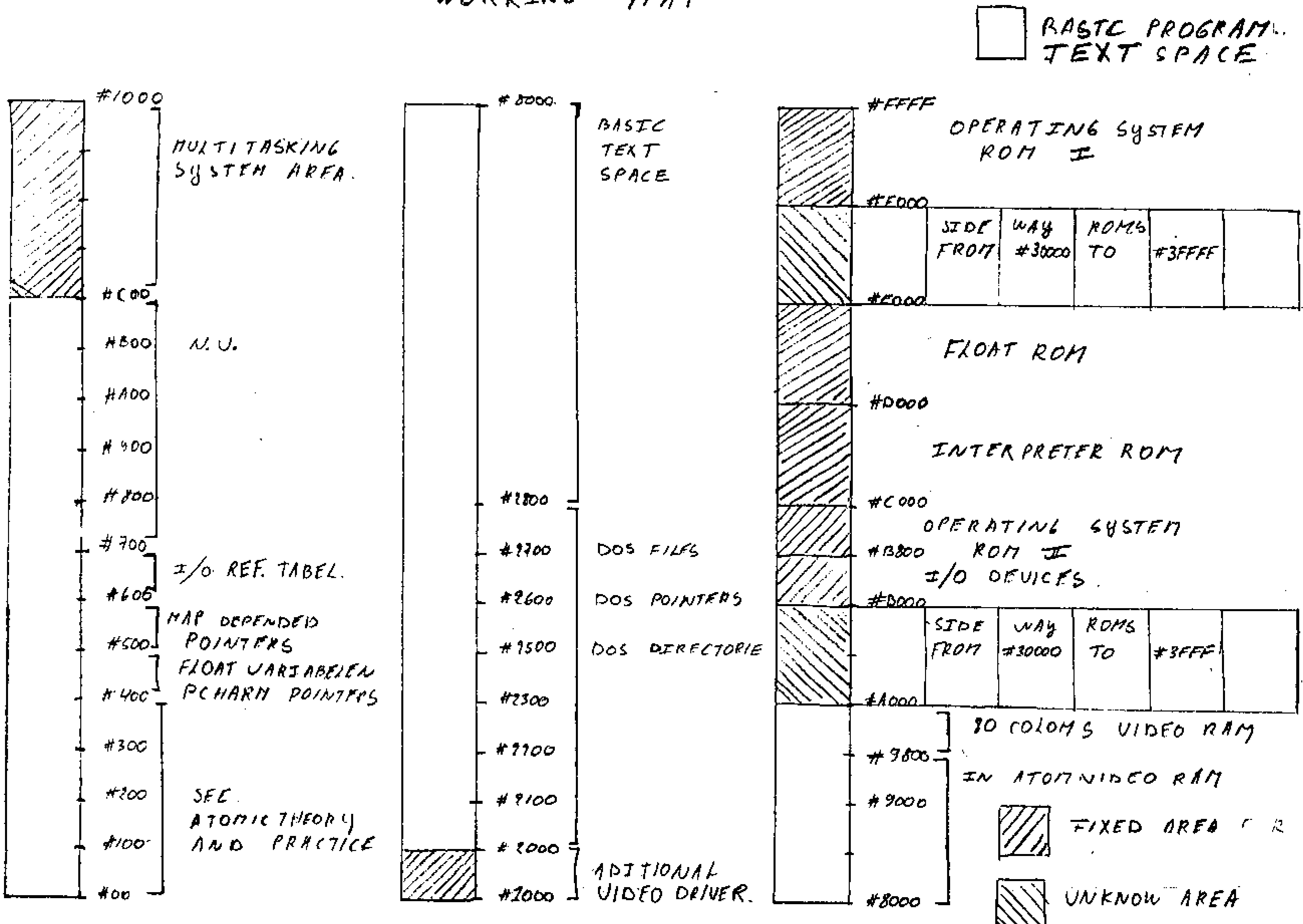


You sure about this new computerized aiming system, Jones?

CURRENT MEMORY MAP



WORKING MAP


BASIC PROGRAM TEXT SPACE
#1000
MULTITASKING SYSTEM AREA.
#C00
#B00
N.U.
#A00
#900
#800
#700
I/O REF. TABEL.
#600
MAP DEPENDED POINTERS
#500
FLOAT VARIABELEN
PCHARN POINTERS
#400
#300
#200
SEE ATOMIC THEORY AND PRACTICE
#100
#00
#8000
BASIC TEXT SPACE
#2800
#2700
DOS FILES
#2600
DOS POINTERS
#2500
DOS DIRECTORIE
#2300
#2200
#2100
#2000
ADITIONAL VIDEO DRIVER.
#2000
#FFFF
OPERATING SYSTEM ROM I
#F000
SIDE
FROM
WAY
#30000
ROMS
TO
#3FFFF
#E000
FLOAT ROM
#D000
INTERPRETER ROM
#C000
OPERATING SYSTEM ROM II
#B800
I/O DEVICES
#B000
SIDE
FROM
WAY
#30000
ROMS
TO
#3FFF
#A000
80 COLOMS VIDEO RAM
#9800
IN ATOM VIDEO RAM
#9000
#8000
FIXED AREA
UNKNOW AREA

BUS STRUCTUUR VOOR DE ATOM ACORN.

Zoals beloofd in het vorige club nummer en op de laatste vergadering gaan we hier verder met onze ATOM te verbinden aan het rack.
Hiervoor moeten 4 onderdelen gebouwd en een modificatie uitgevoerd worden.

Zoals te zien is op fig1 bestaat de verbinding tussen onze Atom en het rack uit een kaartje,2 flatkabels en een kaart die zich in het rack bevindt.
Het eerste kaartje bevat een 64 polige vrouwelijke connector die op P7 van onze Atom past,een 27 tal weerstandjes,die serie geschakeld de lijnaanpassing vormen voor de flat kabels.
Er wordt gebruik gemaakt van 2 kabels met 34 geleiders,dit om de zaak te vereenvoudigen en goedkoop te houden(dure 64 pin connectors).
In beide kabels zijn de onpare geleiders aan de aarde verbonden,dit om de lijnimpedantie konstant te houden en om een afscherming te vormen tussen de signalen onderling.
Zodoende wordt 1 kabel uitsluitend gebruikt voor adrestransport naar het rack;terwijl de andere de data- en controlsignalen van en naar het rack bevat.
Noteer ook even dat alleen de hoogst nodige signalen worden verstuurd,de anderen zullen op de print in het rack terug opgebouwd worden.
Hoe de signalen op de kabel komen is te zien op fig2 en behoeft verder geen uitleg.

Fig3 laat ons de modificaties zien die nodig zijn in de Atom om het geheel te laten werken.
Even een korte uitleg.
Een inverter 74ls04 of 74ls14 wordt bijgeplaatst in de omgeving van het luidsprekertje en verbonden met dunne draadjes.
Haal IC6 en IC23 uit het voetje,plooi de pennetjes 4 en 5 van IC6 en pennetje 5 van IC23 naar buiten en plaats deze terug.Doe hetzelfde met IC6 pennetje 4.
Verbindt de beide pennetjes 5 van IC6 en IC23 met elkaar en verbindt ze dan met de toegevoegde inverter pen 4
Verbindt pen 3 van de toegevoegde inverter aan PL7/31A,verbindt ook het naar buiten geplooide pennetje van IC6 hieraan.
Plaats nu nog een pullupweerstand van PL7/31A naar de plus 5V.
Breng een draadje aan tussen IC8 pin 6 en pin 1 van de toegevoegde inverter,verbindt de uitgang pen 2 hiervan met IC6 pen 4.

Deze voorziening maakt het mogelijk zonder verdere uitdecodering in de Atom,de gebieden #B004 to #B3FF en van #B810 tot #BBFF vrij te gebruiken in het rack.
Dit wordt verder uitgewerkt in het rack door de lijnen BOARD 0 of 1 laag te brengen wanneer een van deze adressen geselecteerd worden.
Op dezelfde manier kan gewerkt worden om de roms en rams uit te schakelen.

In het rack vinden we 2 connecters terug CAx,CBx en Csxx.waar "xx" het pen en "s" de rijaanduiding is.

Noteer hier ook even dat de onpare pennen van CA en CB verbonden worden aan de aarde van rack.

Connecter CA bevat de adreslijnen,terwijl connecter CB de data- en controllijnen bevat.

Even een korte beschrijving van wat de ICs doen op dit bord.
IC1,IC2 en IC4 zijn de respectievelijke adresbuffers naar het rack.Zij kunnen in tristate mode gebracht worden door de signalen NMAP en NDMAR.
IC3 bevat de databuffers die omgeschakeld worden in directie door de gebufferde R/W,en in tristate mode zijn wanneer F2 of NBA laag zijn.

IC4:uit het voorafgaande blijkt dat deze reeds 2 adreslijnen bevat:A16 en A17 die IC1 en IC2 volgen.
De overblijvende buffers worden respectievelijk gebruikt voor R/W en NWDS,NRDS die in tristate mode kunnen zijn wanneer NBA laag is.
IC5 is de buffer voor F1,F2,SYNC,NBA en NRST.

NRST word opgewekt met transistor TR1 en een schmittrigger IC wanneer de voedingsspanning opkomt.Dit signaal gaat naar alle ICs die een "power on reset" nodig hebben.

IC5:maakt het signaal NBA op aanvraag van NDMAR en brengt de respectievelijke busdrivers op het gepaste tijstip in tristate mode.

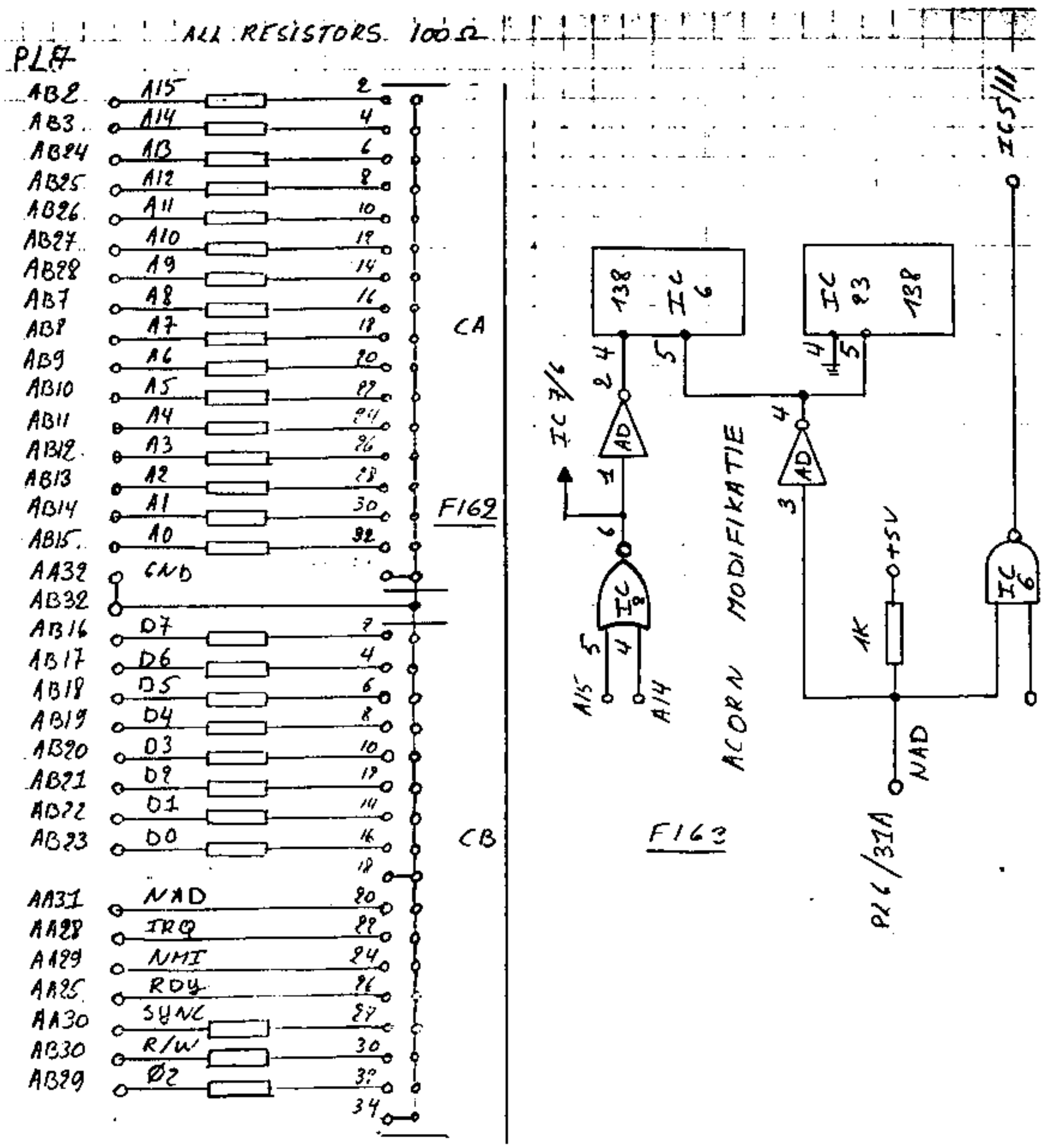
IC7:zorgt dat wanneer NBOARD 0 of 1 laag worden,de Atom uitgeschakeld wordt door de NAD lijn.

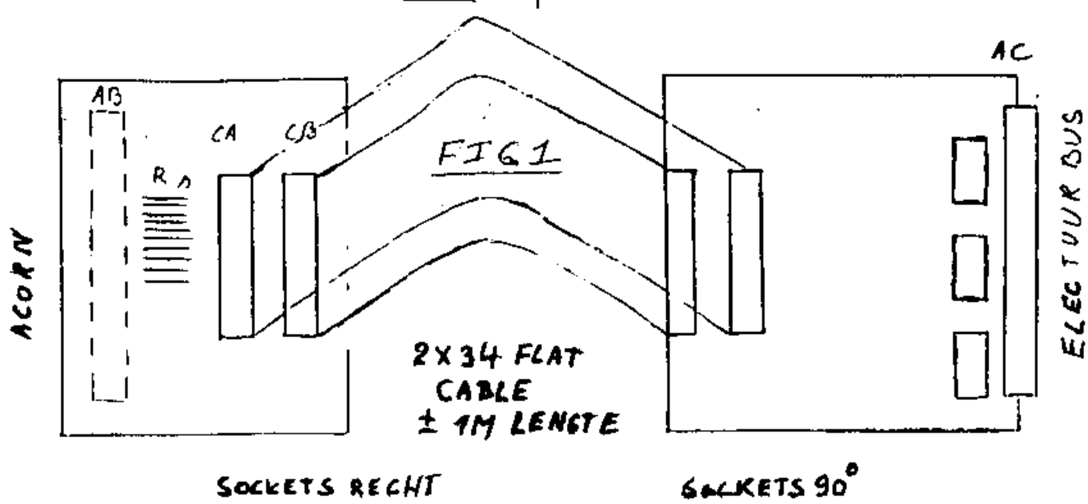
Verdere informatie kan bekomen worden op de volgende club vergadering,tot zolang.

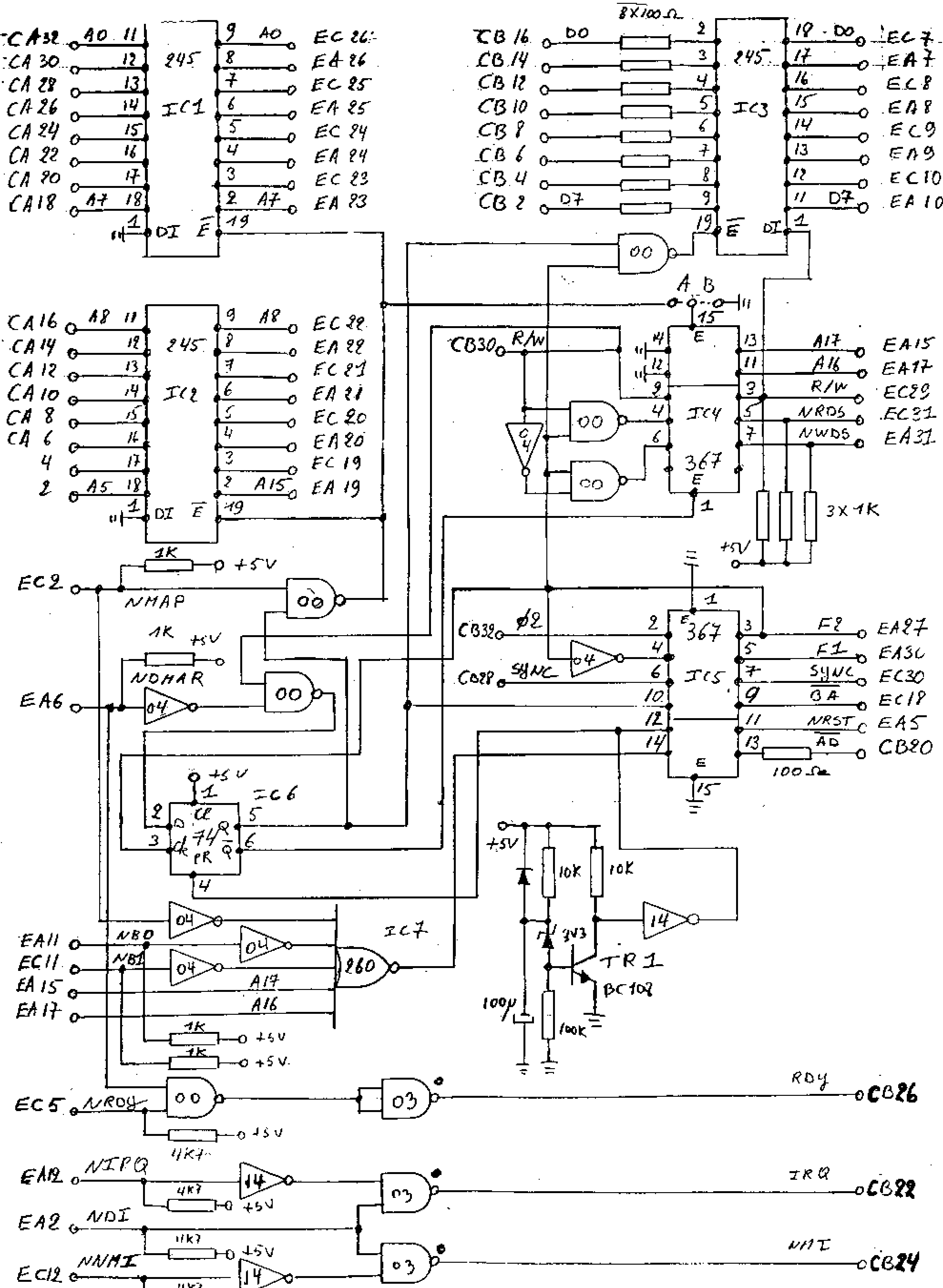
De gemodifieerde Electuur bus.

A		C
+5V	o--1--o	+5V
NDI	o 2 o	NMAP
-12V	o--3--o	-12V
GND	o--4--o	GND
NRST	o 5 o	NRDY
NDMAR	o 6 o	NDMAG
D1	o 7 o	D0
D3	o 8 o	D2
D5	o 9 o	D4
D7	o 10 o	D6
NBOARD0	o 11 o	NBOARD1
NIRQ	o 12 o	NNMI
NDMAR0	o 13 o	NDMAR1
NDMAR2	o 14 o	NDMAR3
A17	o 15 o	N.U.
GND	o-16-o	GND
A16	o 17 o	+12V
-5V	o 18 o	NBA
A15	o 19 o	A14
A13	o 20 o	A12
A11	o 21 o	A10
A9	o 22 o	A8
A7	o 23 o	A6
A5	o 24 o	A4
A3	o 25 o	A2
A1	o 26 o	A0
F2	o 27 o	DMAA
N.U.	o 28 o	DMAB
N.U.	o 29 o	R/W
F1	o 30 o	SYNC
NWDS	o 31 o	NRDS
GND	o-32-o	GND

Note:alle lijnen beginnend met de letter N zoals NDI geven aan dat deze laag actief zijn.


ALL RESISTORS 100Ω
PL7
CA
CB
FIG2
IC 7/6
IC 6
138
IC 23
138
IC 8
IC 6
IC5/111
ACORN MODIFIKATIE
1K
+5V
NAD
PL6/31A
A15
A14
FIG3


ACORN
AB
CA
CB
FIG1
AC
ELECTUUR BUS
2 X 34 FLAT
CABLE
± 1M LENGTE
SOCKETS RECHT
SOCKETS 90°



IC1
245
IC2
IC3
IC4
367
IC5
IC6
IC7
8X100Ω
3X1K
TR1
BC108
NMAP
NDMAR
NRDY
NIRQ
NDI
NNMI
RDY
IRQ
NMI
CB26
CB22
CB24

FIG 4

DOS CONTRLOLLER PROBLEMEN

Toen ik mijn disc drive kocht zowat 3 jaar geleden, heb ik mijn controller gebouwd naar het scheme van TELEC, echter met wat toegevoegde verbeteringen.
Een van de eerste verbeteringen aan deze controller was de ready timing, die wel aanwezig is in de originele Atom controller.
De rede hiertoe is,indien spoel files worden gebruikt,en de file start op track 0 deze track steeds beschadigd werd in sector 0 en 1.
Waarom dit steeds gebeurde alleen op track zero is om de volgende reden. De disk head is op track 0 en moet een write operation doen, maar eerst moet de directory gelezen worden, upgedate en terug geschreven worden. Indien de disc head zich nu op een andere track bevind heeft de drive alle tijd om op snelheid te komen, indien dit echter niet het geval is, is het niet denkbeeldig dat de controller reeds begint met schrijven nog voor de disc op de juiste snelheid is. Dit heeft tot gevolg dat een corupte directory op disc geschreven word.
Om dit nu te vermijden heeft Atom in zijn controller een timer ingebouwd die na 0.5 seconden de geselecteerde ready lijn laag brengt waardoor door de controller na de eerst volgende index puls aan de software meld dat de drive ready is om gebriukt te worden.
Door het invoeren van deze timer verdween voorgaande uevel, doch een ander schoonheids foutje melde zich aan.
Telken de drive aangesproken werd, ook al draaide deze nog wachte de software steeds 0.5 seconden alvorens tot actie over te gaan.
De reden hiervoor kon gevonden worden in de software en het controller IC.
Er gebeurt eigenlijk het volgende. Telkens voor de drive aangestuurd wordt, wordt het adres juist gezet in de controller, ook al is dit niet veranderd.
Door dit her adresseren veschijnt een klein pulsje op de moteron lijn die de timer start, ongeveer 40 micro seconden lang.
Dit 40 micro seconden lange pulsje moet geelimineerd worden indien de drive nog draaid.

Hoe werkt het.
Er word gebruik gemaakt van counters inplaats van one-shot ICs, de reden is dat one-shots voor lange perioden zoals hier 0.5 seconden niet al te stabiel zijn.

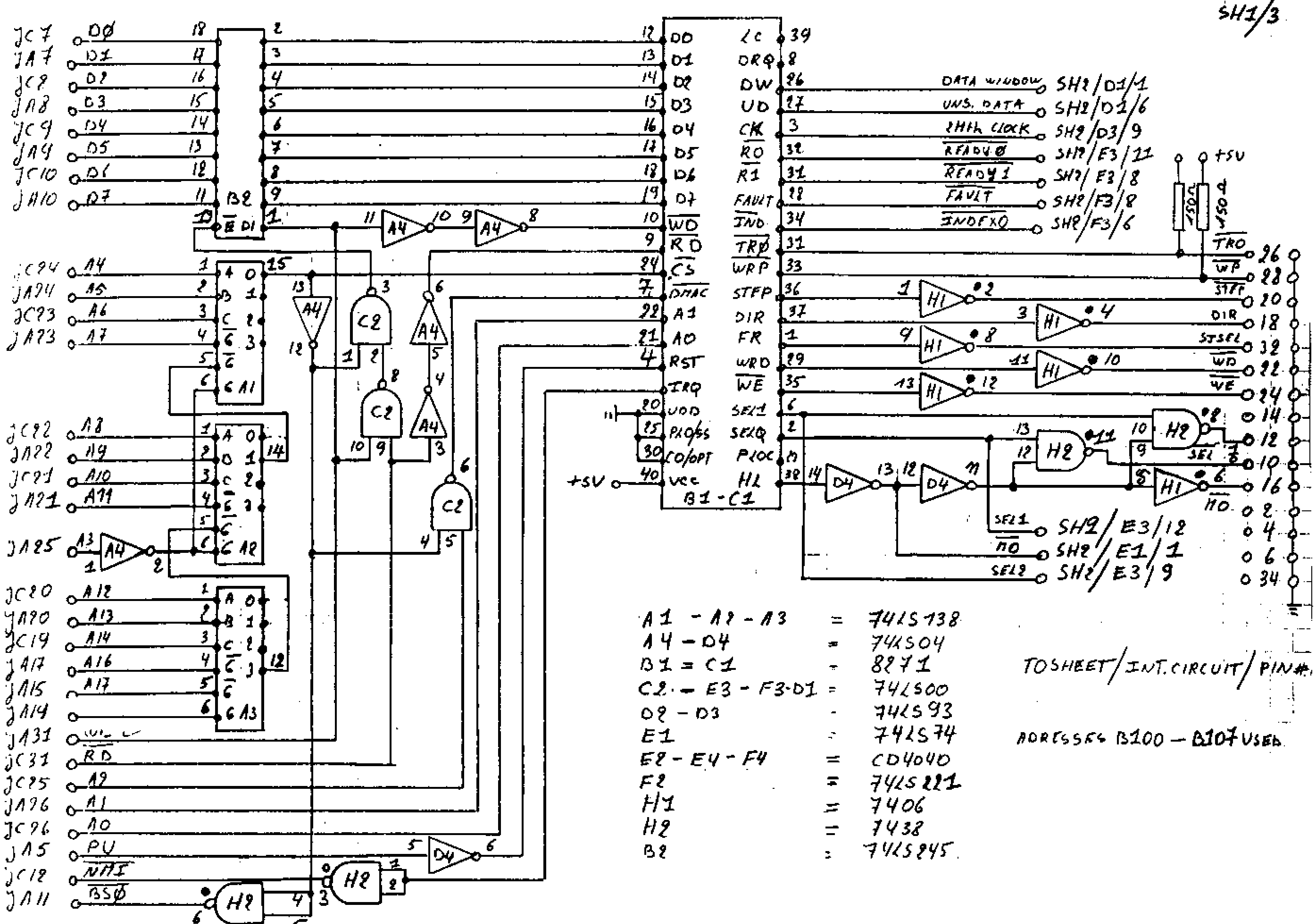
Een 1Mhz clock signaal wordt toegevoegt aan een 12 bit counter IC type CD4040 (E4/10) deze deelt dit door 4096 wat een 244 Hz signaal oplever aan de uitgang van E4/1
Ook word een tweede signaal gekreeerd dat 3906 Hz signaal oplevert, dit stemt overeen met een periode van 255 ms (E4/13).
Deze periode word gebruikt om de40 micro seconden puls weg te werken van de controller.

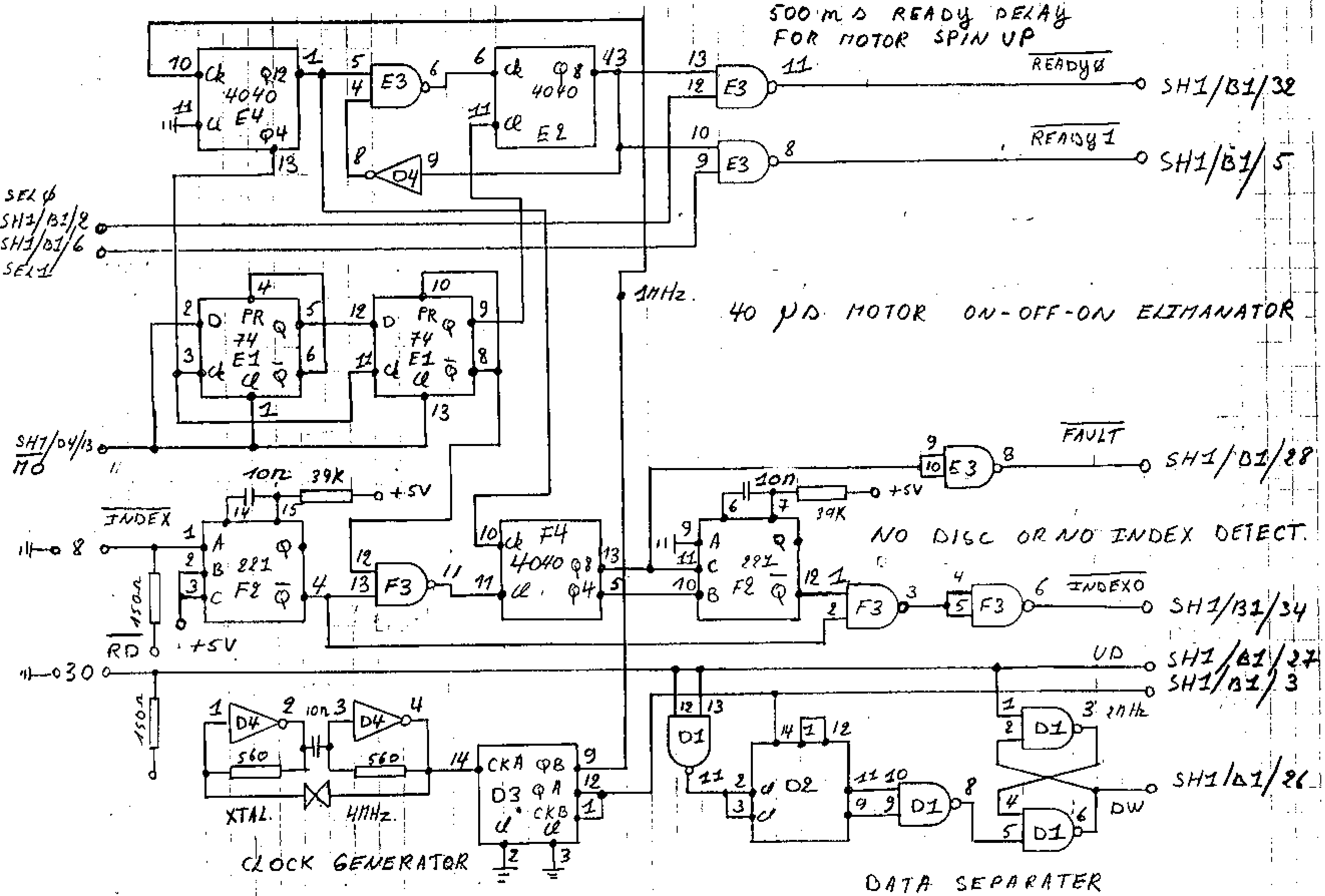
Wanneer de motoron lijn laag wordt, worden de beide flip-flops gereset (E1/1/13) en wordt de reset weg genomen van de counter E2/11, waardoor deze begint te tellen met een klock frequentie van 244Hz wat na 524 ms een hoog oplevert aan de twee nand gates E3/10/13 die zorgen dat de correcte ready lijn laag gebracht

word. Ook word er voor gezorgt dat de teller stop met tellen door de clock pulsen te stoppen die van de counter E4/1 komen.
Hiervoor zorgen de gates D4/9/8 en E3/4/6.
Wanneer nu het motoron signaal hoog word, wat een motor stop van de drive tot gevolg heeft, blijft de hele toestand nog even aangehouden tot de beide flip-flops E1 gekeerd zijn, dit gebeurd ten snelste 250 micro seconden nadat het motoron signaal hoog geworden is, en laatste 2*250 micro seconden later.
Hierdoor word voorkomen dat de timer tussen twee disc accessen in nog zou gestart worden als de drive nog draait.

Een ander probleem stelt zich indien er zich geen disc in de drive bevind of de verkeerde drive geselecteerd is, en een disc komando gegeven wordt.
De disc begint te draaien en er gebeurt niets zelfs geen error melding.
De reden hiervan is niet de software maar een slechte inplimetatie van de hardware, daar de nodige software aanwezig is in het eprom.
Wat is er aan de hand,de controller ziet geen enkele index puls komen, daar de index lijn of hoog of laag blijft, afhankelijk van een open- of gesloten deur, en er zich al dan niet een disc in de drive bevind.
Er is hier echter wel een hardware oplossing voorhanden.
Er weze opgemerkt dat het status register in de controller ingeklockt wordt door een index puls. Dit staat zeer onduidelijk aangegeven in de Intell data sheets van de 8271 controller.
Eerst moeten we van de index lijn een bekent nivau maken, dit doen we door hier een one-shot te plaatsen van 60 micro seconden (F2),die laag actief is.
Deze one-shot geeft een puls door aan de controller langs de nand poort F3/2/3 en de inverter F3/4/5/6
Indien het moter start signaal laag wordt, wordt dit doorgegeven door de flip-flops E1 naar de nand gate F3/12/11, hierdoor wordt het clear signaal weggenomen van de counter F4/11 en deze begint te tellen.
Indien er nu geen index puls komt voor 0.5 seconden gaat deze counter de one shot F2/11 enabelen en volgt er om de 10 ms een puls aan de ingang van F2/10 waardoor deze een dummy index puls genereert van 60 micro seconden die langs de poort F3/1/3 word doorgegeven aan de controller.
Door de actie van F4 word ook langs E3/9/10/8 de fault pin van de controller laag gebracht en is de software nu in staat een DISK ? error te creeren.
Voor verdere informatie kunt u me steeds contakteren op de club vergaderingen.

M.Van Leuven A.C.B


SH1/3
A1 - A2 - A3 = 74LS138
A4 - D4 = 74LS04
B1 = C1 = 8271
C2 - E3 - F3 - D1 = 74LS00
D2 - D3 = 74LS93
E1 = 74LS74
E2 - E4 - F4 = CD4040
F2 = 74LS221
H1 = 7406
H2 = 7438
B2 = 74LS245
TOSHEET/INT.CIRCUIT/PIN#.
ADRESSES B100 - B107 USED.
B1 - C1
B2


500 mS READY DELAY FOR MOTOR SPIN UP
40 µS MOTOR ON-OFF-ON ELIMANATOR
NO DISC OR NO INDEX DETECT.
CLOCK GENERATOR
DATA SEPARATER
READY0
READY1
FAULT
INDEX0
SH1/B1/32
SH1/B1/5
SH1/B1/28
SH1/B1/34
SH1/B1/27
SH1/B1/3
SH1/B1/26
SEL0
SEL1
SH1/B1/2
SH1/B1/6
SH1/04/13
MO
INDEX
RD
4040
E4
E2
E3
D4
74
E1
221
F2
F3
F4
D1
D2
D3
1MHz
+5V
10n
39K
150Ω
560
XTAL.
4MHz.
UD
DW

EEN MAGIC ATOMIC I/O SYSTEEM.

Daar ik in het bezit ben van enkele I/O boards en deze random wil gebruiken, zonder steeds de I/O vectoren #0208 tot #020B te moeten omzetten, ontstond het idee van een flexibel I/O systeem.
Het systeem bestaat uit minimum 2 roms, t.t.z.een rom die het gebied #A000 tot #AFFF beslaat, die basic I/O functies bevat, en een of meerdere roms die het gebied #E000 tot #EFFF beslaan, die een of meerdere i/o drivers en device handlers bevatten.
Daar de I/O basic statements zich in het #AXXX gebied bevinden en toch functies van andere roms in het gebied van #AXXX, moeten kunnen uitprinten is het best te begrijpen dat hiervoor een zeer efficient en schakel sijsteem ontwikkeld werd wat dit toelaat, b.v.b "EPRINT:TRUE"is best mogelijk.
De werking is als volgt : indien data naar het scherm, de printer en de RS232 printer gezonden moet worden, dan moet er eerst verteld worden aan het systeem welk device welk referentie nummer heeft.
Er moet dus eerst een I/O systeem configureerd worden. Dit gebeurd door gebruik te maken van systeem commando's (*commando's). Het systeem is er op gebouwd dat de rom die aangeroepen word geen vectors initializeerd zodat de dos rom niet in de weg gelopen wordt. Dit wil zeggen dat de rom alleen in geschakeld wordt wanneer hij nodig is en de originele rom terug inbrengt wanneer hij verlaten wordt. Er word helemaal geen gebruik gemaakt van de huidige systeem vectoren.
Hiervoor moeten echter wel enkele andere punten bij geschaafdt worden, hierover echter verderop meer.
Terug naar het configureren. Voor het scherm met het bijhoren keybord is dan de systeem handler en device driver "VIDEO" ontwikkeld. Voor de printer werd dit "PRINTER" en voor het RS232 interface werd het RS232L (waar de L staat voor local).
Nu kan aan elk van deze devices, een referens nummer toegewezen worden. Het "DEVICE REFERENS NUMBER" een getal van 0 tot en met 15.
Dit ziet er dan in direct mode or program mode (met line number) als volgt uit :"*VIDEO,0" , "*PRINTER,1" EN "*RS232L,2".
De device referens numbers moeten niet in opeenvolgende volgorde ingegeven worden, Het mogen ook andere getallen zijn.
Wat gebeurt er,indien zo een device geconfigureerd word. Er is plaats voorzien in ram om een tabel op te bouwen die 256 bytes lang is, en elk device neemt daar een plaatsje in van 16 bytes (bij mij bevind zich deze tabel van #0600 to #06FF nog eens 2 maal 16 bytes scratsh path van #05E0 tot #05FF).
Voor elk device worden nu in deze tabel 16 bytes geschreven die het volgende aangeven:
BYTE #8 ,het rom nummer welke de device driver bevat (mag niet nul zijn).
BYTE #0 tot BYTE #7 driver afhankelijke data, waaronder command bytes die gebruikt worden door de driver zelf of die gebruikt worden om iets door te geven aan het I/O statement.
BYTE #9 timeout waarde.
BYTE #A en #B indirekt adres pointer voor de driver input buffer.
BYTE #C en #D indirekt adres pointer voor de driver output buffer.
BYTE #E en #F indirekt adres pointer voor de driver command routines.
Indien er b.v.b. een string moet geprint worden op de printer,

moet niet meer vooraf de printer aangezet worden, dit doet de nieuwe driver zelf, er moet alleen in het print statement verteld worden met welk device er moet gewerkt worden. Om dit eenvoudig te houden werd gekozen om de statement name intakt te houden en enkel een "E" vooraan aan de naam toe te voegen.
In het voorbeeld komt printen op de printer er als volgt uit te zien, "10 EPRINT,1:" zodat eenvoudig kan volstaan door in reeds bestaande programmas de statements op te zoeken en vooraan een "E" toe te voegen en de nodige parameters achter de statement name, wat heel snel kan gebeuren met de P-CHARM editer.
In bovenstaand voorbeeld word dus printen op de printer in "10 EPRINT,1:$A" en op de serial printer "20 EPRINT,2:$A".
Ook werden de nodige errors voorzien,die indien er iets verkeerd moest lopen,zoals een oproep naar een niet geinitializeerde device, (ERROR 3) . Data input van een device dat geen data kan geven geeft ook een error doch dit is de normale data error die voorkomt,omdat het device andwoord met "CR" wat een data error oplevert of een lege string variabele. Data printen naar een input device levert geen error op daar de data verloren gaat in het bit bucked.
Even in het kort wat er gebeurd indien een extended I/O functie word aangeroepen. De interpreter gaat op zoek naar het statement en gaat dit uiteindelijk vinden in een "AXXX" rom. De snelheid waarmede dit gebeurd is afhankelijk van de schakelsoft en is minimum 4 miliseconden zonder schakelen bij een normale ATOM van 1 MHz. Dit heeft voor gevolg dat indien telkens data uitgeprint moet worden die van verschillende functies in het statement terecht komen, dit een tamelijk trage bedoening zou worden. Om dit te ondervangen word de data eerst in een buffer van 256 bytes (bij mij in gelegen in het gebied van #7F00 to #7FFF) geschreven en aan de driver doorgespeeld wanneer de buffer vol is, of indien het statement beindigd wordt. Deze methode laat toe data te tranfereren in burst mode met snelheden groter dan 20000 baud.
Wanneer het statement gevonden is, wordt aan de hand van het referens nummer, dit voor later hergebruik weggezet, de device referens tabel terug gevonden. Deze word gecopieerd naar scrath path tabel 2 om een snelle actie mogelijk te maken, daar de 6502 niet indirekt met ofset kan springen. Het rom nummer wordt opgehaald uit scratch path tabel 2 en voorgschakeld, het ":" teken word geskipt en het interpreteren van het verdere statement kan afgehandeld worden. De data gaat in de buffer en behandeld zoals voorgaandelijk is beschreven. Om af te sluiten word het device referens tabel terug gecopieerd naar zijn oorspronkelijke plaats, het oorspronkelijke #EXXX rom terug voorgeschakeld en het I/O rom verlaten.
De i/o statements en functions, "PRINT FPRINT INPUT FINPUT LIST PUT BPUT SPUT FPUT GET BGET SGET en FGET", werden hiervoor herschreven en kregen voor erkenning een "E" vooraan zodat het statement "PRINT","EPRINT" wordt e.z.v..
Alle i/o statements hebben nu een syntax die er als volgt uitzien:STATEMENT,PAR1,PAR2,PAR3:VARIABELEN.
Indien PAR1 weggelaten wordt dan is deze bij default nul, idien PAR1 en PAR2 nul zijn (optional commandparameters voor de driver), worden de default parameters gebruikt die in de device tabel staan.
Een statement kan er dus als volgt uitzien "ELIST,A,B,C:D,E,F"
Waar A het device refferens number is ,B en C optional driver parameter zijn (hierover later meer).

Het dubbel punt teken, vormt de scheiding tussen de driver parameters en de te behandelen variabelen.
Tijdens het herschrijven werden enkele belangrijke verbeteringen aan de i/o statements aangebracht als volgd:

ELIST:
======

Dit statement krijgt een belangerijke parameter meer, namelijk de tekst-space-pointer.
Het ziet er dan als volgt uit: "ELIST:A,B,C",waar "B" en "C" de vroegere parameters waren, en "A" nu de tekst-space-pointer voorsteld, indien deze vervangen word door een comma, dan word de tekst-space-pointer die in #12 staat gebruikt.
Verder kan het statement nu ook nog gebruikt worden in programmas daar waar dit vroeger alleen in direckt mode bruikbaar was.
Hiermede kan nu een programma geschreven worden dat een na een andere programmasvan disk haald en op de printer list, zelfs kan een programma zichzelf listen.

EINPUT en EFINPUT:
==================

Hier zijn enkele belangrijke wijzegingen aangebracht.
V.B.het steeds in de weg zittend vraagteken is verdwenen, en indien u het terug wenst moet u het zelf zetten.
Een groot voordeel is echter in de plaats gekomen. Daar alle input gebufferd is , leest het statetement de benodigde data uit de buffer, en gebruikt hiervoor de pointers "INUSE" en "INCOUNT". Aan de hand hiervan weet het statemen waar de volgende data zich in de buffer bevindt.
Even een voorbeeld. Er moeten 3 parameters ingelezen worden,namelijk een variabele "A",een string "$B",en een variabele "C".
In het "INPUT" statement ging dit als volgt:
10 INPUT "A",A,"$B",$B,"C",C
Wat ons het volgende beeld opleverde:
A? 10 cr
$A? HELLO cr
B? 20 cr
Nog een voorbeeld:
20 INPUT"A,$B,C",A,$B,C
A,$B,C? 10 cr
? HELLO cr
? 20 cr
Met het nieuwe "EINPUT" en "EFINPUT" is dit veel mooier te verwezeliyken.
10 EINPUT:"A",A,"$B",$B,"C",C
Hier is het normale antwoord zoals in het voorgaande statement mogelijk en ziet er dan uit als volgt:
A 20 cr
$B HELLO cr
C 30 cr
Ook kan als volgt geantwoord worden:
A 20,HELLO,30 cr ik weet wat er gevraagt word dus geef ik het antwoord al maar,
$BC de vraag komt er later echter no wel uit.
Het voorgaande voorbeeld naar lijn 20 levert dan het volgende op:

20 EINPUT:"A,$B,C",A,$B,C
A,$B,C 10,HELLO,20 cr
Of ook nog als antwoord:
10,HELLO cr
20 cr
Hieruit valt op te merken dat zowel de comma als de cr als separeter kan gebruikt worden. Een additionele toevoeging aan EFINPUT is dat deze nu ook strings kan binnen lezen.

EPRINT en EFPRINT:
==================

Hier zijn geen ingrijpende veranderingen gebeurd. Alleen wordt de data die uitgeprint wordt in een buffer van 256 bytes geschreven en het is deze buffer die doorgegeven wordt aan de driver. Ook kan EFPRINT strings afdrukken wat met de normale FPRINT niet mogelijk was.

EPUT,EBPUT,ESPUT:
=================

Hier is alleen het file argument vervallen daar zich dit in de driver parameters bevind.
syntax EPUT,1:A of ESPUT,1:$A

EGET,EBGET en ESGET:
====================

Hier kan hetzelfde gezegt worden als in het voorgaande behalve voor "ESGET" dat nu de algemene function opbouw volgt en waar de parameter nu aan de linkse zijde staat van het gelijkheids teken.
syntax A=EGET,1: of $A=ESGET,1:

EAPUT en EAGET:
===============

dit zijn nieuwe statements speciaal ontwikkeld om block data transfers toe passen;hiermede kan een blok data van een bepaalde plaats in het geheugen met een maximum lengte van 256 bytes naar een I/O device doorgestuurd worden of er van gelezen worden en op een bepaalde plaats in het geheugen gezet worden.
Zij zijn vooral bedoeld om communicatie tussen verschillende systemen toe te laten.
syntax EAPUT,1:A,B

BUFFER,INCOUNT EN INUSE:
========================

Dit zijn nieuwe funfties die ingevoerd werden om gemakelijk de data in de buffers te kunnen behandelen.
syntax A=BUFFER

EFORMAT:
========

Dit statement laat formatted i/o toe in zowel intiger- als in floating point format.Hiermede kunnen het aantal decimalen voor en na de comma gespecifieerd worden, die moeten afgedrukt woerden, alsook het niveau waarop de exponentiele waardemoet aanvangen waaneer het getal kleiner is dan 1.
syntax EFORMAT,A,B,C

DSATUS0 en DSTATUS1:
====================

Dezen geven in de aangegeven intiger variabelen de device statusbytes 0 en 1 terug, waarin feitelijk dstatus0 een coppy is van het laatste commando aan de driver plus het laatste error type is, en dstatus1 het meest bijzondere deel van van de controller status bevat.
syntax A=DSTATUS0,1

CSTATUS0 en CSTATUS1:
=====================

Dezen functies geven in hun respectivelijke integer variabelen elks 4 controller status bytes terug, die volledig afhankelijk zijn van het gebruikte controller ic.
syntax A=CSTATUS0,1

BUFFERIN en BUFFEROUT:
======================

Dezen statements en functies verzetten de defauld buffer adressen naar een opgegeven waarde of geven deze terug in de hun aangegeven intiger variabelen.
syntax A=BUFFERIN,1 of BUFFERIN,1,A.

SUMCHECK:
=========

Deze functie geeft in de geassosieerde intiger variabele de checsom terug die berekend is over het totaal aantal bytes die uitgegaan of binnengekomen zijn in het laatste i/o commando.
syntax A=SUMCHECK

Tot hier een klein intermetzo van wat zoal moet mogelijk zijn met dit atomic I/O systeem.Gedeeltelijk is het reeds werkende en indien mogelijk zal dit op een van de volgende vergaderingen gedemonstreerd worden.
Een van de problemen die zich nog voor doen is de schakel kaart van de klub die niet toelaat op dit ogenblik van "#AXXX en #EXXX" roms door elkaar te gebruiken en de non-inteligentie van de meeste schakel systemen die ik tot nu toe heb mogen bewonderen in de diverse klubbladen.
Maar geen nood indien we met ons allen de koppen eens bij elkaar steken en wat eigenwijsheid en egocentrisme laten varen komt dit nog wel in orde.

Tot op heden werd de volgende software geschreven en is onder test.

** 1 ** Het A rom, de I/O statements bevattende en ongeveer 4K byte lang.

** 2 ** Een E rom bevattende de volgende drivers:

*PRINTER:
=========

Deze bevat een driver routine die gebruik maakt van de normale in de atom geinstalleerde via 6522.
Het grote verschil met de originele driver is dat deze binaire waarden aankan. Daar een timeout is ingevoerd, programmas niet nodeloos meer blijven hangen, indien de printer niet aanstaat.

*DISPLPR:
==========
Deze driver werd speciaal ingevoerd om programmas die de printer aanzetten door middel van P.$2; deze data te kunnen dirigeren naar eender welk device..Het is eigenlijk een tussen driver en verwijst steeds naar een reeds geconfigureerd device.

*VDU45:
=======
Dit is een 80 coloms video driver gebasseerd op de 80 colloms video kaart van electuur.

*VDU47:
=======
Dit is de normale video driver in de ATOM.

*GRF47:
=======
De normale grafics van de ATOM worden hier gebruikt, er wordt alleen het scherm omgeschakeld.

*VIDEO:
=======
Deze driver gebruikt de voorgeschakelde device handlers *VDU45 en *VDU47 met het keybord zoals zij op dat moment zijn voorgeschakeld.

*PPRINTER:
==========

Dit is een nieuwe volledig op centronics interface gebasseerde driver die 2 verschillende printers over dezelfde interface kan aansturen en ook hun respectivelijke statussen kan interpreteren.

*MEMORY:
========
Deze driver schrijft of leest data in memory van of naar variabelen. Dit is een psuedo random file systeem.

*DELDEV:
========
Dit commando delete een device in de device reference tabel.

Deze drivers zijn bij elkaar ongeveer 4K byte lang.

** 3 ** Een E rom bevattende de volgende driver.

*RS232L
========
Dit is een volledig lokale RS232 interface driver gebasseerd op de 6551.
Deze is echter alleen geschikt om lokale devices aan te sturen zoals plotters,line printers, terminals en het overpompen van data uit een andere computer in een dos inspool file. Dit is reeds onder verschillende omstandigheden met verschillend devices getest zoals:
HP 1000 real time computer sytem RTE 6 VM.

HP 7274 6 pen plotter.
HP 2934A graffics line printer.
COMMODORE C64 home computer.

Deze driver kan de volgend handschake protocols aan:
1 hardwired handshake (DSR-DCD-DTR-RTS-CTS).
2 XON - XOFF ofte DEC protocol.
3 ENQ - ACK ofte HP protocol.
Deze RS232L driver is ongeveer 2K byte lang.

Verder moet nog gezegt worden dat geen enkele van deze drivers interupts gebruikt. Daar ik van oordeel ben dat de ATOM structuur helemaal niet geschikt is om meerdere interupts tegelijk te kunnen verwerken. De reden hiervoor kan gezocht worden in de 6502. Deze bezit geen uitwendige status lijn die kan aangeven dat hij een interupt aan het afhandelen is ja dan neen.
Dit zou nog kunnen hardwire matig kunnen opgelost worden, ware het niet dat reeds enkele software packetten geschreven zijn die het RTI statement gebruiken om naar een bepaald adres te springen,ik noem hier maar PCHARM en GAGS.
Moeten verder nog geschreven worden, een IEE488 en een RS232M (modem) driver.



Keep trying, Lefty . . . we still have exactly seven minutes 'till the cops get here! . . .

MACORN.

Macorn wordt een pakket van programma's voor de Acorn Atom. Met het hoofdprogramma, ge-ent op het programma 'Macdraw' van Apple, kan men op eenvoudige wijze tekenen met behulp van een joystick. Het gehele pakket heeft het volgende nodig:
Een Acorn Atom met geheugen van #2200-#3FFF en #8000-#9800
Een diskdrive
Een GAGS-ROM versie 3.0E
Een joystick.
Dus, in tegenstelling tot het programma 'MACINTOSH' van Dick van den Ende, heeft MACORN geen schakelsoft nodig. Macorn heeft meer mogelijkheden dan het programma MACINTOSH, is minder gestructu- reerd en minder snel.
LET OP:
RENUMBer het programma MACMAIN nooit! Daarin wordt namelijk ge- spronGen naar een (REGELNUMMER+(10*VARIABELE)). Het programma zou dan vast lopen. Ik geef toe dat dit genoeg spaghetti ople- vert om 6 hongerwinters te overleven, maar het bespaart wel ruimte.

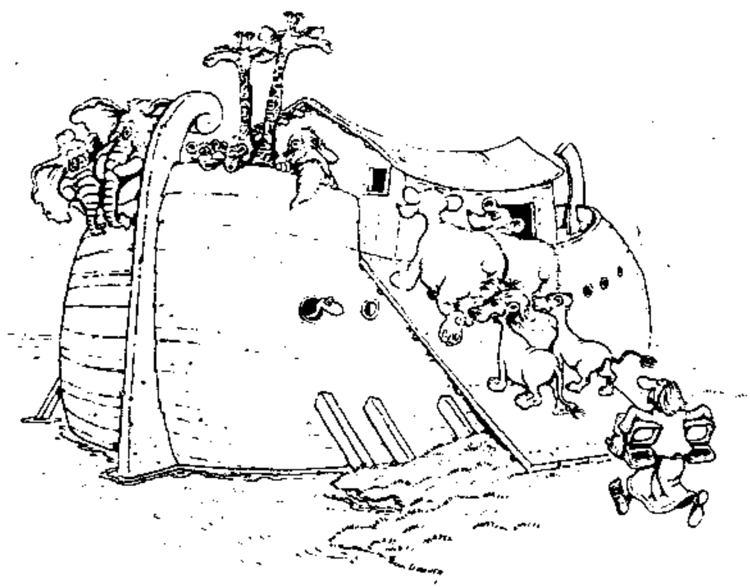
De mogelijkheden.
Als u het programma opstart met
*MACORN (RETURN)
dan krijgt u het 'configuratiescherm' te zien. Druk als u uitge- keken bent op een toets en het MACORN-logo verschijnt in beeld. Wacht tot de diskdrive uitgedraaid is en u kunt tekenen.
Boven in het scherm ziet u 16 paintpatronen. In de linkerboven- hoek zit een zwart vlakje. Als u het kruisje van het midden van het beeld daarop zet en de vuurknop indrukt, dan krijgt u de an- dere 16 patronen te zien.
Het programma MACORN werkt gewoon met het middelste bit van het kruisje. Als u met het kruisje op zo'n vakje met een patroon gaat staan en u drukt de vuurknop in, dan kiest u zo voor dat patroon. Als u nu het kruisje tussen de twee balken brengt en de vuurknop weer indrukt, wordt het zwarte vlak met het door u ge- kozen patroon gevuld.
In de linker- en rechterbenedenhoek ziet u, geinverteerd, res- pectievelijk 'C' en 'DS' staan. Die staan daar om de boel een beetje op te vullen en hebben dus geen andere betekenis dan 'COPYRIGHT' (op zich al onzinnig) en 'Dice Soft'. Deze naam geeft al aan hoe dit programma gemaakt is: op de gok.
Naast de 'C' staat weer een zwart vakje. Als u voor dat commando kiest (op de zelfde manier als dat u een paintpatroon kiest) dan krijgt u de andere 12 commando's te zien. We bespreken eerst de balk met commando's die u het eerst in beeld ziet.
Vam links naar rechts :
(?) Help. Geef een beknopte uitleg van de commando's. U begrijpt er waarschijnlijk niets van als u eerst niet dit heeft gelezen. Help is dus alleen maar 'n geheugensteuntje. (W) Wegschrijven. Het plaatje wordt van #8000-#9800 geSAVEd.

(L) Laden. Een plaatje op programma wordt op #8000 geladen.
(I) Inverteer scherm. Het scherm wordt d.m.v. een BASIC routine geinverteerd. Ik heb het statement INV expres niet gebruikt omdat de paintpatronen dan ook geinverteerd worden, wat verwarrend kan werken.
(C) CX-System. Hieronder moet een aantal extra commando's wor- den geimplementeerd, dat alleen draait met een goede scha- kelsoft.
(*) *CAT. Geef een *CAT van de schijf in de drive.
(S) Schoon. Wis het scherm.
(A) Een commando dat ook niet gedefinieerd is, bij gebrek aan goede ideeen. Suggesties zijn welkom (bedreigingen niet).
(P) Piep. Geeft een pieptoon bij de keuze van een commando.
(M) Macorn. Een beknopt overzicht van alle sub-programma's. Is wel gedefinieerd, de programma's zijn er echter nog niet.
(T) Tekst. Hiermee kunt u tekst neerzetten. De breedte, hoogte, dikte en plotmode van de letters zijn door u te definieren. (Q) Quit. Duidelijk, neem ik aan.

De betekenis van de tekencommando's is (van links naar rechts):
1 Lijn. Trek een lijn met de beweging van de joystick mee.
2 Lijn .tot. . Trek een lijn van punt tot punt.
3 Stippellijn. Trek een stippellijn. De lengte van de stippen zijn door u te definieren.
4 Frame. Zet een vierkant c.q. rechthoek op het scherm.
5 Stuf. Stuf iets op het scherm uit.
6 Blok. Zet een gevuld blok op het scherm.
7 Cirkel. Zet een cirkel op het scherm.
8 Printerdump. Maakt een printerdump van het scherm. Die moet u zelf inbouwen. De regels waaraan u zich moet houden staan verderop beschreven.
9 Inkt. Definieer de plotmodes van alle statements.
10 Driehoek. Zet een gelijkbenige driehoek op het scherm.
11 Acorn. Zet een acorn op scerm.
12 Copyright. Zet een copyright-teken op het scherm.
Tijdens het tekenen van een figuur hoeft u de vuurknop niet in- gedrukt te houden, daar wordt u alleen maar moe van. De routine begint als u de vuurknop indrukt en stopt als de knop weer wordt ingedrukt. Hij begint weer bij de volgende druk op de knop.
De invoer vanaf het toetsenbord wordt soms niet gedefinieerd. Dan zou ik wat extra routines moeten schrijven en daar wordt IK moe van.

Het installeren van een Printerdumproutine.
In regel 350 van het programma MACMAIN staat als eerste commando G.a.. Dit commando zorgt voor het terugspringen naar het bestu- rings gedeelte. Dit moet u dus verwijderen. Hierna moet u de printerdump laden. Als u geen geheugen vrij meer heeft, kunt u dit

```
 10 REM TEST ACORN-ATOM
 20 REM TOTAAL TESTPROGRAMMA IS SAMEN GESTELD
 30 REM UIT HET PROGRAMMA TEST AUTEUR ?????
 40 REM EN HET PROGRAMMA CRC TEST UIT A.N.
 50 REM JAARGANG 2 NO:4 AUTEUR LEENDERT
 60 REM SAMEN "GEPLAKT" EN AAN GEVULD DOOR
 70 REM  H.J.SPIERENBURG
 80 REM  ALARDUSDREEF 57
 90 REM  3235VB ROCKANJE
100 P.$12
110 P=#2800;P.$21
120 P.$2"TEST ACORN-ATOM"',$3,$21
130[
140LDY@0;LDA(#80),Y;STA#82;LDA@#EC;STA(#80),Y;LDA(#80),Y
150CMP@#EC;BNEP+14
160LDA#82;STA(#80),Y;INY;BNEP-19;LDA@0;STA#80;RTS
170LDA@1;STA#80;RTS
180];P.$6;@=0
190 P.$15
200!#90=#ABCDEF01;P.$2'"ZERO PAGE: ";IF!#90=#ABCDEF01;P."O.K."'
210IF!#90<>#ABCDEF01;P."DAMAGED"'
220P=#1000
230DO;!#80=P;LINK#2800;P=P+#100;U.?#80=0
240P."#"&P-#100" - #"
250DO;!#80=P;LINK#2800;P=P+#100;U.?#80=1 O. P=#2800
260P.&P-1" RAM O.K."'
270P."#2800 - #31FF THIS PROGRAM"'
280P=#3200;DO!#80=P;LINK#2800;P=P+#100;U.?#80=1
290P."#3200 - #"&P-#101" RAM O.K."'
300 P.$3
310 REM CRC TEST
320 DIM LL3,A(12),B(5)
330 RESTORE 820;Z=0;@=0
340 FOR N=0 TO 3;LL(N)=-1;N.N
350 FOR N=0 TO 1;P.$21
360 P=#2800;[
370        LDA @#10
380        STA #32
390        LDY @#00
400        STY #30
410        STY #3A
420        STY #3B
430:LL0 LDA (#30),Y
440       LDX @#08
450:LL1 LSR A
460       ROL #3A
470       ROL #3B
480       BCC LL2
490       PHA
500       LDA #3A
510       EOR @#2D
520       STA #3A
530       PLA
540:LL2 DEX
550       BNE LL1
560       INY
570       BNE LL0
580       INC #31
```



```
 590        DEC #32
 600        BNE LL0
 610        RTS
 620+;N.N;P.$6
 630 P.$2
 640 P."  ROM/     BEREKENDE  WERKELIJKE"'
 650 P." EPROM     SIGNATURE  SIGNATURE"'
 660 F.Q=0TO 11
 670 ?#BFFF=7
 680 READ $A,P,$B
 690 ?#BFFF=Z
 700 ?#31=P
 710 LINK#2800
 720 C=?#3B*#100+?#3A
 730 P.$A," #",&C,"         ",$B'
 740 IF Q=3 THEN P." A-BLOKKEN"';Z=0
 750 IF Q)3;?#BFFF=Z ;Z=Z+1
 760 NEXT Q
 770 P.$3
 780 REM DATA"SOORT ROM",ADRES,"SIGNATURE"
 790 REM DATA EVENTUEEL AANPASSEN
 800 REM AAN EIGEN SYSTEEM
 810 REM VANAF REGEL 780 (A-BLOKKEN)
 820 DATA"BASIC LOW  ",#C0,"#DE7D"
 830 DATA"BASIC HIGH ",#F0,"#C3BA"
 840 DATA"DOS ROM    ",#E0,"#4C45"
 850 DATA"F.P.ROM    ",#D0,"#3069"
 860 DATA"EP.0 AXR-1 ",#A0,"#C1B5"
 870 DATA"EP.1 DISK  ",#A0,"#CB6F"
 880 DATA"CX-2.7     ",#A0,"#FAD4"
 890 DATA"LEEG       ",#A0," ????"
 900 DATA"CX-3.02    ",#A0,"#ED71"
 910 DATA"EDITER     ",#A0,"#B326"
 920 DATA"GAGS       ",#A0,"#4198"
 930 DATA"P.CHARME   ",#A0,"#C16B"
 940END
>
```

```
● RUN
TEST ACORN-ATOM
ZERO PAGE: O.K.
#1000 - #27FF RAM O.K.
#2800 - #31FF THIS PROGRAM
#3200 - #9FFF RAM O.K.
  ROM/      BEREKENDE  WERKELIJKE
 EPROM      SIGNATURE  SIGNATURE
BASIC LOW    #D67D      #D67D
BASIC HIGH   #E386      #C3BA
DOS ROM      #C20E      #4C45
F.P.ROM      #3069      #3069
 A-BLOKKEN
EP.0 AXR-1   #9263      #C1B5
EP.1 DISK    #2A84      #CB6F
F.P.ROM      #3069      #3069
EP.0 AXR-1   #4198      #C1B5
EP.1 DISK    #B0F1      #CB6F
CX-2.7       #5B16      #FAD4
LEEG         #54D1       ????
CX-3.02      #2C9F      #ED71
```

op #2000 of #2200 doen, zolang de dumproutine maar niet bo- ven #2900 komt. Daarna kunt u uw programma aanroepen. Als het programma klaar is moet het programma terugspringen naar regel 355 van MACMAIN, waar MACSET weer op #2200 geladen wordt. Als u een geheugenkaart heeft en de Printerdump niet op #2200-#2900 laadt, kunt u deze regel er uit halen en direct springen naar label 'a'; dit gebeurt ook in regel 356.

Voor vragen en/of suggesties:
J.Imthorn
Lindestraat 12
3297 XM Binnenmaas.

```
*I.
   HUIS    8000 8000  01800 002
   IK      8000 8000  01800 01A
   MACLETT 2900 CE86  007E5 032
   MACMAIN 2900 CE86  01192 03A
   MACALL  2E00 CE86  005DC 04C
   readme  2800 A071  01585 052
   MACINKS 2900 CE86  0014C 068
   MACINK  8000 8000  00200 06A
   MHELP4S 2900 CE86  00340 06C
   MHELP3S 2900 CE86  00213 070
   MHELP2S 2900 CE86  00247 073
   MHELP1S 2900 CE86  00239 076
   MHELP3  8000 8000  00200 079
   MHELP2  8000 8000  00200 07B
   MHELP1  8000 8000  00200 07D
   MACWRT  2600 CE86  002DD 07F
   MACDIGI 2200 CE86  00380 082
   MACSET  2200 CE86  00600 086
   MACPAT  2900 CE86  0057A 08C
A  ATOMTST 2900 2900  008D6 092
   MCONFIG 8000 8000  01800 09B
   MACLOGO 8000 8000  01800 0B3
   MACORN  2800 CE86  0011F 0CB
```

Nadat we de vorige keer de SR-flipflop uitgebouwd hebben tot een RAM-chip, zullen we deze aflevering weer in de sfeer van de geheugens houden (het mijne is toch al zo slecht). We zullen een aantal verschillende vormen van komputergeheugen bekijken. Komputer geheugen is in principe in 2 delen te splitsen:

- werkgeheugen
- achtergrondgeheugen.

Werkgeheugen RAM, ROM etc.

In deel 2 hebben we geheugen in een heel eenvoudige vorm gezien: de SR-flipflop (latch). Een relatief eenvoudig te begrijpen geheel, met op zich zeer nuttige toepassingen (dendervrije schakelaar, geheugencel). Alleen is het als gebruikersgeheugen niet zeer spektakulair in deze vorm. Daarvoor moeten we een aantal flipflop's bij elkaar nemen en er een RAM-chip van maken.

Zo'n RAM-chip heeft een aantal voordelen, zoals een vrije toegankelijkheid om te lezen en om te schrijven, een lage access time (=aanspreektijd). Een nadeel van RAM is, dat het vluchtig is. Dit betekent, dat als de voedingsspanning verdwijnt, de inhoud van het geheugen ook verdwenen is. Dit kan vervelend zijn, vooral als het informatie betreft, welke vaak gebruikt wordt.
Nu lopen er gelukkig veel slimme mensen rond, zodat het niemand zal verbazen, dat er op een goede dag een RAM-chip gemaakt is, welke door een paar batterijtjes op spanning gehouden kan worden (dit is mogelijk in de zgn CMOS techniek). Een bekend voorbeeld is de binnen onze klub veel gebruikte 6116. Deze chip wordt echter buiten de hobbysfeer ook veel in de professionele wereld toegepast. Het stroomverbruik is zo laag, dat 2 batterijtjes het maanden en maanden uithouden.
Let wel, dit geheugen is in principe ook vluchtig.

In principe niet vluchtig is een ROM. ROM staat voor Read Only Memory. Het kan dus alleen gelezen worden. Een ROM kenmerkt zich meestal door een zeer lage access time. Een ROM is dus zeer snel aanspreekbaar.
IC 20 van de ATOM is een ROM. Het bevat het besturingsprogramma (operating system), de BASIC interpreter en de assembler.
Een ROM heeft bij de fabrikage bepaalde informatie meegekregen, welke zich niet laat veranderen.

Een broertje van de ROM is de PROM (Programmeble Read Only Memory). Dit geheugen is eenmalig door de gebruiker te programmeren (geen fouten maken bij het programmeren dus). Ook een PROM heeft een lage access time.

Voor de hobbyisten interessanter is de EPROM (Erasable PROM). Dus een PROM, welke te wissen is, en wel met UV licht (vandaar het venstertje op de chip). Nadeel is dat het programmeren een aparte bezigheid is, welke los staat van het normale gebruik. Het programmeren vindt veelal in een speciale programmer

plaats. Het programmeren is een vrij langdurig proces: 50 ms per byte. Dit geeft bijvoorbeeld een programmeertijd van 3.5 minuten voor een 4 kByte EPROM (bijv. 2532). Ook een EPROM heeft veelal nog een lage access time (alhoewel je snelle en langzame typen hebt). Een relatief nieuwe ontwikkeling is het Inteligent Programming Algorithm van Intel. Hiermee kan een speciaal gebouwde EPROM in een fraktie van de gewone tijd geprogrammeerd worden.

Verdere vormen: RePROM = Reprogrammable PROM (vergelijk EPROM).

EAROM = Electrically Alterable ROM. Kan in de schakeling geprogrammeerd en gewist worden, maar dit is weer een langzaam proces. De EAROM kan dus niet als RAM gebruikt worden.

Kenmerkend voor al deze ROM's is dat ze vrij weinig stroom vragen. Verder als groot voordeel: als de komputer uitgaat, blijft de DATA aanwezig. Het geheugen is niet vluchtig (non-volatile).

Achtergrongeheugen band disc etc.

Een andere vorm van geheugen is het achtergrondgeheugen (ook wel back-up geheugen, het opslag medium of extern geheugen).

Een voor alle Atomisten bekend voorbeeld is de kassetterekorder met de kompaktkassette. De access time van de kassette recorder is erg groot (heen en weer spoelen, programma zoeken en inladen). Voordeel is de prijs, welke erg laag kan zijn.
De signaaloverdracht is vaak niet erg groot (1200 Baud = bits/s), maar deze is voor ons dankzij een slimme klubgenoot uit te breiden tot 9600 Baud. Hiermee kan op een simpele C60 kassette een schat aan programmatuur vastgelegd worden.

Voor wie dit allemaal niet snel genoeg gaat, of voor diegene die meer wil dan zijn buurman, is er de floppy disc. Dit opslagmedium, welke voor het eerst in 1962 door IBM op de markt gebracht werd, is het volgende station voor vele kassetterekorder gebruikers. Het is veel professioneler. Want wordt de floppy disc niet gekenschetst door een behoorlijk lage access time, bedraagt de opslagkapaciteit niet tussen de 100k en 1M byte per schijfje, is de bit-overdracht van floppy naar komputer niet flitsend (we praten nu niet over een C'64 uiteraard) en werkt het systeem niet nagenoeg probleemloos, nadat het bij het begin van de ontwikkelingen in de jaren 50 een flop was? Dan wil iedereen toch wel zo'n pragramma-opslagapparaat of niet dan? Hopelijk dat recente ontwikkelingen in het Oosten van het land spoedig de weg naar de huiskamer weten te vinden.

Gaan we nog een stapje verder, dan komen we bij de hard disc (ook wel Winchester disc, naar een ontwikkeling van IBM). Deze hard disc wordt over het algemeen gevormd door een aluminium schijf, met een magnetische laag erop. Deze schijf heeft een vaste plaats in de behuizing van de drive. Dit is nodig, omdat het geheel luchtdicht afgesloten moet zijn. Vlak boven het oppervlak van de schijf, welke met zo'n 3600 omw/min

ronddraait, bevindt zich nl de lees/schrijf kop. Deze hangt er zo dicht boven, dat er nog geen stofje tussen kan.
De floppy disc werkt ook volgens dit principe (de schijf wordt in sporen (koncentrische cirkels) en segmenten verdeeld). Maar bij de hard disc gaat alles wat sneller en is alles wat gevoeliger. Vandaar de extra voorzorgsmaatregelen.
De opslagkapaciteit van de hard disc varieert van 10 tot ruim 100 Mbyte. Dit samen met een lage access time en een zeer hoge bit-transmissie van en naar de komputer vormen de voornaamste features van de hard disc. Als veiligheids back-up voor floppies en hard-discs wordt vaak nog weer een groot magneetband geheugen gebruikt. Dit hoeft niet supersnel te zijn, omdat het geen werk kopieen zijn. Mocht er dus onverhoopt toch iets fout gaan in de floppy of de hard disc, dan is de programmatuur toch niet verloren.
Geruchten gaan dat de hard disc ook al hier en daar door hobbiisten gebruikt wordt (tja, al die spelletjes passen natuurlijk nooit meer op een floppy ..).

Waar dat naar toe moet gaan? Een blik op de toekomst.

In de (nabije) toekomst schijnt het optische geheugen zijn intrede te gaan doen, met kreten als CD-ROM en laser-straal.
De opslagcapaciteit is gigantisch, terwijl access tijden lijken te vervagen. Het systeem is gebaseerd op de compact-disc speler en kan dientengevolge slechts als Read Only Memory dienst doen. Systemen voor in de auto met plattegronden etc. zijn al aangekondigd.
De laser-straal van een compact-disc speler kan het plaatje alleen aftasten (lezen). Ontwikkelingen zijn er reeds gaande (bijv. bij Philips), om systemen te realiseren, welke ook op het schijfje kunnen schrijven (PROM). Ja, zelfs waarbij de aangebrachtte programma's ook weer gewist kunnen worden (EPROM). Dit schrijven en wissen stellen weer specifieke eisen aan de laser-straal (20x zo hoge intensiteit als bij de CD) en aan het materiaal van het plaatje. Wissen geschiedt hierbij ook met de laser-straal, zodat dit selektief plaats kan vinden.
Welke van deze versies het uiteindelijk zal worden en hoe groot de opslagkapaciteit zal worden, zal wellicht niet zozeer door de techniek, als wel door de commercie bepaald worden.

Alhoewel er meer opslagmedia zijn (zie tabellen) heb ik met de hier gegeven opsomming de belangrijkste en meest gebruikte willen noemen.

De cijfertjes nog eens op een rijtje.

Bekijken we tenslotte nog even de opslagkapaciteiten van een aantal verschillende opslagmedia (de getallen zijn eerder een indikatie dan een keihard gegeven).

medium	aantal tekens		A4 equivalent
1 vel papier .	16	kbit	1
halfgeleidergeheugen	256	kbit	16
bellengeheugen	1	Mbit	62,5
magnetische schijf	560	Mbit	35000
magnetische band	720	Mbit	45000
cassette C60 analoog	860	Mbit *	62500
LP analoog	1,2	Gbit *	75000
holografisch geheugen	10	Gbit	630000
compact disc	15	Gbit	940000
optische plaat 30 cm (io)	20	Gbit	1300000
magneto-optische plaat 30 cm (in ontwikkeling)	30	Gbit	1900000
videoband analoog	150	Gbit *	9400000
videoplaat analoog	150	Gbit *	9400000
menselijk geheugen	1000	Gbit	62500000

*) digitale waarde, berekend uit de analoge waarde.

Opgemerkt zij verder nog dat 1 byte = 8 bit.

Het menselijk geheugen blijkt dus nog steeds superieur. Zou het allemaal dan toch nog wel een beetje meevallen met dat geheugen van mij??

Logische groeten,
Wibo Visser.

CORRECTIE van: PROGRAM VERTRAGING.
Zie ATOM NIEUWS Jr.5 No.2 Pag.73

```
10 PROGRAM VERTRAGING
20 DIM VV2
30 P=#322E;[
40      LDY @#10
50:VV0 LDA #337
60:VV1 LDX @#FF
70:VV2 DEX
80      BNE VV2
90      SEC;SBC @#01
100     BNE VV1
110     DEY
120     BNE VV0
130     RTS;]
140 END.
```

Een wel zeer goedkoop alternatief voor de gewone mini schakelkaart.

De 2532 EPROM, die in onze computer in het voetje van IC 24 past, wordt ten opzichte van de 27XX serie EPROM'S steeds duurder. We kunnen al een 2764 EPROM kopen, die een kapaciteit heeft van 8K, voor ongeveer 1/3 van de prijs die velen van ons in IC 24 voetje hebben zitten.

Er is echter een probleem. In het 2732-voetje kunnen we niet zondermeer een 27XX serie EPROM zetten. De pinconfiguratie van de 27XX serie wijkt namelijk af van de 25XX serie. Om hieraan iets te doen moeten we een tussenvoet maken, die het voetje geschikt maakt voor een 2732 (die ook al aanmerkelijk goedkoper is als een 2532).

HET MAKEN VAN EEN VERLOOPVOET.

Wanneer we de aansluitingen van de 2732 naast die van de 2532 leggen, zien we dat er maar een paar aansluitingen verschillen. Dit zijn de pennen Vpp, A11 en CE.
Om nu de 2532 voet geschikt te maken voor een 2732, hebben we 2 IC voetjes nodig (liefst met gedraaide contacten), die we recht boven elkaar monteren, en de pennen 1 t/m 17 en 19 en 22 t/m 24 van beide voetjes doorverbinden.
Nu de verschillen:

van 2732 voet (pin)	naar	2532 voet (pin)
18	-	20
20	-	21
21	-	18

Zie ook figuur A.
Wanneer we nu deze tussenvoet in het IC24 voetje steken, kunnen we in de tussenvoet een 2732 gebruiken (ipv 2532).
N.B. Deze tussenvoet kunnen we ook gebruiken om de 2732 te programmeren, in een 2532 programmer (clubprogrammer).

LET OP ! er bestaat een 2732A, die in tegenstelling tot de normale 2732 een programmeerspanning heeft van 21V ipv 25V nodig heeft. !!!

Het zou leuk zijn, als we ook ee 2764 in dit voetje kunnen prikken, waar we dat 2 utility EPROMS in onder kunnen brengen, die we dat eenvoudig met een schakelaartje om kunnen schakelen.

De 2764 heeft 4 pootjes meer dan een 2732, maar verder is de pinconfiguratie op de Vpp pen na gelijk. Hiervan kunnen we gebruik maken.
Wanneer we nu een 28 pens voetje (liefst gedraaide contacten) nemen en de pennen 1, 2, 27 en 28 afknippen, dan past dit voetje precies in het zojuist gemaakte tussenvoetje voor de 2732.
Voor de omschakeling tussen de twee 4K banks, waarin 2 UTILITY EPROMS zijn geprogrammeerd (bijv. P-CHARME en GAGS) moeten we een schakelaar aanbrengen tussen de massa en A12 resp. pen 14 en pen 2. Wanneer we een enkelpolige schakelaar willen gebruiken, moet er tussen pen 27 en 2 een pull up weerstand (10K) gemonteerd worden. Verder verbinden we de pennen 1, 28, 27 en 26 met elkaar. Zie figuur B.
Nu kan er in het zojuist gemaakte voetje een 2764 geprikt worden, en dit geheel gaat weer in de tussenvoet, en dat dit alles weer in het voetje van IC 24 in de computer.

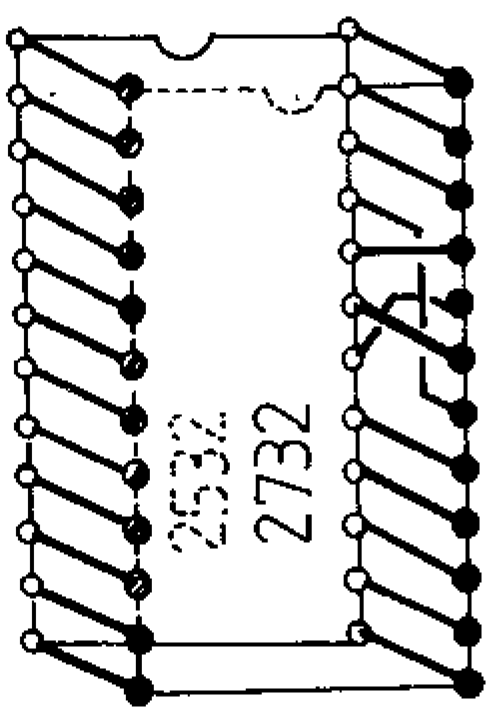
NB Met deze voet kan GEEN 2764 geprogammeerd worden. Dit zal in een aangepaste programmer moeten gebeuren (bijv het nieuwe model ZERO programmer.)

Wanneer de schakelaar gesloten is, dan zal de lage bank van 4K in gebruik zijn, bij een open schakelaar gebruiken we de hoogse 4K van de EPROM.
Als we pen 2 van het voetje aan een uitgang van de VIA i.p.v. de schakelaar hangen, kunnen we banks (BOXEN) softwarematig omschakelen.

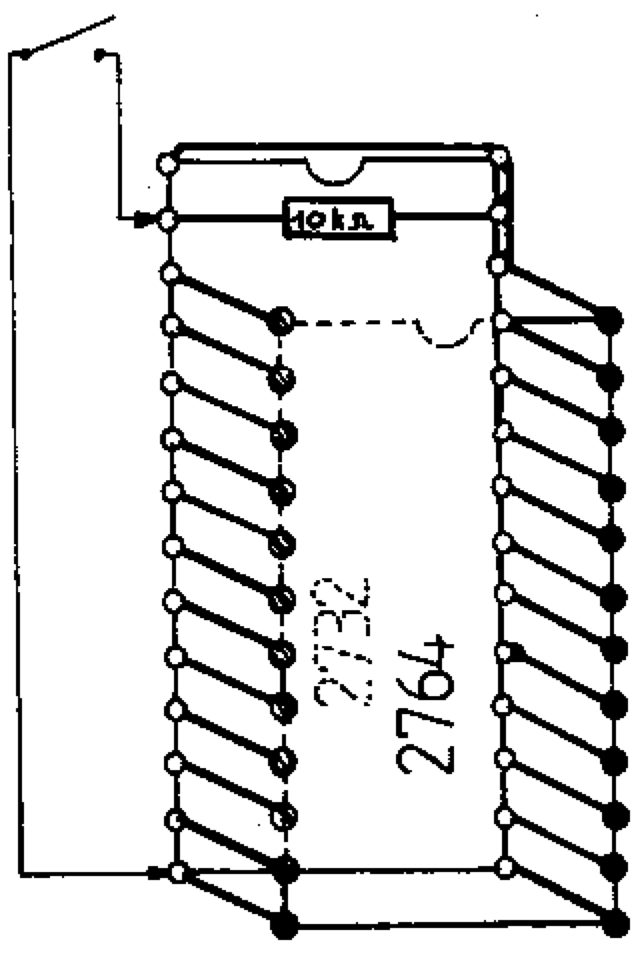
Deze truuk is in principe bruikbaar voor alle 4K 2532 EPROM's in onze computer (FP, DOS).
Wanneer uw schakelkaart al is omgebouwd voor 2732 EPROMS dan is de eerste tussenvoet natuurlijk niet nodig.

Veel succes !!

Rudi van Drunen.



a



b

Zoals U wellicht van Uw regio-opperhoofden gehoord zult hebben is er een nieuwe hardwarecommissie gekozen op de Algemene Leden Vergadering te Utrecht.
Op dit moment is deze hardwarecommissie (HWC) nog druk in de slag om bepaalde voor haar belangrijke beleidslijnen vast te leggen; er zijn echter uit de eerste vergadering reeds enkele punten naar voren gekomen. Deze punten zullen in dit artikeltje behandelt worden.
Allereerst een oproep aan alle leden, en alle regionale HWC's. Als U iets experimenteerd in Uw Atom, hoe klein of simpel het ook is; zet hiervan alstublieft een schema op papier, schrijf alstublieft een korte beschrijving, en stuur dit op aan de HWC. (Adres staat op pagina 1)
Vaak komen er zaken naar voren die reeds lange tijd bij iemand in de Atom draaien, en die wellicht zeer interessant zijn voor meerdere leden. Zonde, want we hadden hier misschien eerder gebruik van kunnen maken, en de HWC had U hier misschien in kunnen ondersteunen... Insturen dus!!
Over de punten die naar voren zijn gekomen in de eerste vergadering het volgende:

Er is een stevige ondersteuning van de reeds bestaande standaardisatie "wensen" van de vorige HWC. Als we allemaal op een bepaalde plek iets anders plaatsen wordt het een puinhoop, niemand kan iets van een ander meer lezen.
De HWC is dan ook vast van plan om deze standaardisatie uit te breiden, maar ook vooral nogmaals te wijzen op het "gevaarlijk" programmeren door misbruik van monitor routines. (Zie artikel van Bram Poot over deze materie)
Verder zal er gekeken worden of we ons volledig aan de door Acorn meegeleverde memorymap houden, of dat we hier nog iets in wijzigen (bijvoorbeeld de Acorn Floppy Disk Controller).

Proffesionaliteit moet het devies worden van de huidige HWC, het leek ons zeer belangrijk dat projecten, die voor iedereen worden ontwikkeld "af" zijn. M.a.w. fouten zoals die er m.n. met de schakelkaart zijn geweest mogen niet meer voorkomen. Nico Stad heeft in het vorige nummer al wat hierover aangehaald met het stukje over de 80-kolomskaart. Er worden eerst door een aantal mensen prototypen ontwikkeld en gebouwd, als deze geheel werken wordt er een handleiding opgebouwd, en als deze klaar is wordt er nog een prototype door een leek op hardwaregebied gebouwd. Als dit proces zonder problemen verloopt is het project klaar voor productie. Deze methode kost meer tijd maar het is heel wat waard om een kaart probleemloos te laten werken......
Wat enigszinds betrekking heeft op bovenstaande; bij iedere kaart die uitgebracht wordt dient software "meegeleverd" te worden; afhankelijk van de kaart wordt dat een "basis-soft", of een "standaard-soft". Als voorbeeld: met de 80-kolomskaart valt heel wat te experimenteren, hierbij wordt dus alleen een "basis-soft" geleverd; maar als we naar de Floppy Disc Controller gaan kijken zien we dat deze soft thuis hoort op het #EXXX-blok,

verder zien we dat Gerrit Hillebrand reeds een volledige soft voor de kaart geschreven heeft; een "standaard-soft" dus.

Misschien heeft U het reeds gehoord; er worden bij nieuwe kaarten pakketjes met onderdelen geleverd. Deze methode is door het bestuur gekozen omdat het U erg veel in de knip gaat schelen, -de federatie betrekt de zaken bij enkele groothandels- en, met deze methode kunnen er wat kwaliteitseisen gesteld worden. Mijn stokpaardje is dat U ten alle tijden gebruik moet maken van IC-sockets met gedraaide contacten om contactproblemen te voorkomen. Wel, dit soort zaken horen thuis in het hoekje kwaliteit, en doordat we met grote hoeveelheden inkopen hebben we voor een zeer interessante prijs gedraaide sockets; deze zijn dus in de -in antistatisch materiaal verpakte- pakketjes terug te vinden. Om U in ieder geval wat vergelijkend materiaal te geven voor wat betreft het 80-koloms pakket zoals dat binnenkort geleverd gaat worden, de richtprijs die we in het hoofd hebben voor de kaart is ƒ 130.--, dit is dus een pakket met alle onderdelen -behalve RAM's, hierover later meer-, de print, en een setje DIN41612 connectors...... (Alles op deze kaart wordt geleverd in High Speed CMos)
Als laatste van de reeks punten komt het "oppoetsen van oude hardware", we moeten ons hierover nog een duidelijk beeld vormen, maar een indicatie is dat er een nieuwe Atom-Ware komt, met werkelijk alle denkbare informatie van alle releases van de schakel- en geheugenkaart. Men zou het een Atom wie-wat-en Ware kunnen noemen.
De eerste prioriteit van de HWC is het coordineren van de nodige hardware-praktijken, hierdoor hopen we te zorgen dat niet 10 mensen tegelijkertijd het wiel (dat onderling weer enigszinds verschilt) uitvinden. We zijn tevens een instantie waar regionale HWC's bij grote problemen met projecten op terug kunnen vallen. We hopen in ieder geval de zittingsperiode van deze HWC soepel te laten verlopen, en te zorgen voor een wat groter scala van uitbreidingsmogelijkheden van ons geliefde, en soms verfoeide, computertje.

Tot zover de eerste levensverschijnselen van de nieuwe HWC, en nogmaals, laat iets van U zien.....

Yvo Tuk

Met dit programma kunt u u biorithmen berekenen. Volgens de leer van de biorithme wordt ons leven beinvloed door 3 cycli, die starten op de dag van geboorte. Deze cycli verlopen sinusvormig. Tijdens de positieve helft voelen we ons goed, tijdens de negatieve helft zouden we voorzichtig moeten zijn. Als kritieke dagen gelden de dagen, waarbij de sinus de nullijn passeert.

Ten eerste de fysische cyclus van 23 dagen. Deze beinvloed onze conditie, ons uithoudingsvermogen, en de weerstand tegen ziekten.
De tweede is de emotionele cyclus van 28 dagen, welke ons gevoel, de emoties en onze intuitie beinvloed.
De derde is de intellectuele cyclus van 33 dagen, welke onze helderheid van geest beinvloed.

```
   10 PROGRAM BIORYTM
   20 REM BY M.A. CLEGG
   30
   60 P.$12"***BIORYTHM***"'"-------------"'
   70 DIM M13,X10
   80 !M=#1F1C1F00; M!4=#1F1E1F1E;
M!8=#1E1F1E1F;M!12=#1F
   90 P."IS YOUR NAME ON FILE (Y/N)"'
  100 DO INKEY Z;U.Z=CH"Y" OR Z=CH"N"
  110 IF Z=CH"Y"; GOS.a
  120bP.'"TYPE NUMERIC DATE OF BIRTH"'
  130 IN."DAY "A,"MONTH"B,"YEAR",C
  140 IF C<1900; C=C+1900
  150 D=0;IF C%4 AND B>2; D=1
  160 Z=0;IF C%4=0 AND B=2; Z=1
  170 IF B<1 OR B>12 G.d
  180 IF A<1 OR A>(M?B)+Z; G.d
  190 G.c
  240dP."INVALID DATE, TRY AGAIN"';G.b
  250cP.$12;@=0;P."OK, BORN ON:"A"-"B"-"C'
  255gIN."IN WHICH MONTH DO YOU"'"REQUIRE BIORYTHMS"E
  260 IN."...AND THE YEAR"F
  270 IF F<1900; F=F+1900
  280 IF E<1 OR E>12 OR F<1981 OR F>2100 G.f
  290 G.e
  300fP."INVALID DATE, TRY AGAIN"';G.g
  310eG=0;H=0;@=0;I=0
  320 REM CALCULATE DAY BIRTH TO MONTH
  330 F.Z=1 TO B; G=G+M?(Z-1);N.;G=G+A
  340 F.Z=1 TO E; H=H+M?(Z-1);N.
  350 IF D=1; G=G+1; REM G= DAY IN BIRTH YEAR
  360 IF F%4=0 AND E>1; I=1
  370 IF I=1; H=H+1
  380 J=((F-C)*365)+H-G+((F-C)/4)
  390 REM CALCULATE BIO DAYS
  400 K=22-J%23;L=27-J%28;N=32-J%33
  401 P.'"RESULTS ON PRINTER (Y/N)"'
  402 DO INKEY W;U.W=CH"Y" OR W=CH"N"
  403 IF W=CH"Y" P.$2
```

```
  410 P.'"THE CRITICAL DAYS FOR"'"DATE OF BIRTH
"A"-"B"-"C
  415 IF W=CH"Y";P." IN "E"-"F
  420 P." ARE: "'
  430 P."physical ";IF K-12)0;P.(K-12)" AND "
  440 P.K;IFK+11(M?E+I; P." AND "(K+11)
  450 IFK+23(M?E+I; P." ALSO "(K+23)
  460 P.''"emotional ";IF L-14)0;P.(L-14)" AND "
  470 P.L;IFL+14(M?E+I;P." AND "(L+14)
  480 IF L+28(M?E+I;P." ALSO "(K+23)
  490 P.''"intellectual ";IF N-17)0;P.(N-17)" AND "
  500 P.N;IF N+16(M?E+I;P." ALSO "N+16
  510 P.''"TO DISPLAY GRAPH PRESS SPACE BAR"
  515 P.$3
  520 INKEY A
  530 REM GRAPH BASELINES & LETTERS
  540 CLEAR4; COLOUR1
  550 PLOT 12,0,0;PLOT 5,0,192
  560 PLOT 12,0,96;PLOT 5,128,96
  570 DIM R50; Y=3
  580 REM NO. & LETTER VECTOR R
  590
!R=#6E3E4477;R!4=#0E1B6B4D;R!8=#1C494C19;R!12=#00000028
  600
R!16=
00000000;R!20=#47172A3B;R!24=#00286633;R!28=#0000000
  610
R!32=#33660000;R!36=#13467417;R!40=#194C2800;R!44=#0808

  620 F.X=1TO((M?E)+I);PLOT
12,(4*X),96;PLOT1,0,-2;PLOT1,0,4
  630
IFX%5=0;PLOT1,0,2;PLOT1,0,-8;PLOT0,-3,-4;D=X/10;GOS.z
  640 IFX%5=0;PLOT0,2,0;D=X%10;GOS.z
  650 N.X
  660 PLOT12,5,4;Y=3;D=5
  670 F.W=0TO17;D=D+1;GOS.z;D=D+1;GOS.z;PLOT0,2,0;N.W
  680 PLOT12,28,4;PLOT1,14,0
  690 COLOUR2;PLOT12,70,4;PLOT1,14,0;PLOT12,114,4
  700 COLOUR1
  710 F.W=0TO14;U=0;IFW%3=0;U=1
  720 PLOT(U),1,0;N.W
  730 PLOT12,4,16;D=44;GOS.z;D=45;GOS.z
  740 PLOT12,4,184;D=42;GOS.z;D=43;GOS.z
  750 REM DRAW PHY CURVE
  760 PLOT 12,-1,96
  770 F.X=-(2+4*K)TO((M?E)+I)*4-2-4*K
  780 S=X+2+4*K
  790 Y=%(80*SIN(0.0683229*X))
  800 PLOT 5,S,(Y+96);N.X
  810 REM DRAW EMO. CURVE
  820 PLOT 12,-1,96;COLOUR2
  830 F.X=-(2+4*L)TO((M?E)+I)*4-2-4*L
  840 S=X+2+4*L
  850 Y=%(70*SIN(0.0561224*X))
  860 V=12;IFS)1; V=5
  870 PLOTV,S,(Y+96);N.X
  880 REM DRAW INT. CURVE
```

```
 890 PLOT 12,-1,96;COLOUR 1
 900 F.X=-(2+4*N)TO((M?E)+I)*4-2-4*N
 910 S=X+2+4*N
 920 Y=%(60*SIN(0.047619*X))
 930 V=12;IFS%3=1; V=5
 940 PLOTV,S,(Y+96);N.X
 950 INKEYA;P.$12$10$10$10$10
 960 P."DO YOU WISH TO CONTINUE"'"(Y/N)"
 970 DO INKEY A; U.A=CH"Y" OR A=CH"N"
 975 IF A=CH"Y"; RUN
 980 P.'"END      BYE";E.
 990zREM SUB PLOT LETTER
1000 Q=R?D;F.Z=1TO7
1010 A=(2-Z%6)%2;B=(2-(Z-1)%4)%2
1020 PLOT(Q&1),(B*2),(A*Y)
1030 Q=Q/2;N.Z
1040 RETURN
2000aREM STANDARD DETAIL SUB.
2010 IN."YOUR INITIALS PLEASE"$X
2020 IF $X="MC"A=21;B=4;C=1959;G.c; REM DAY,MONTH,YEAR

2021 if $x="PK" A=15;B=7;C=1962;G.c; REM: 15-07-1962
2035 IF $X=""; G.b
2040 P."INITIALS NOT RECOGNISED"';G.a
```



Na al geruime tijd op zoek te zijn geweest naar een geschikte plaats voor de FDC (Floppy Disc Controller), zijn we uiteindelijk terecht gekomen op de adressen #EFF8 tot #EFFF. Oorspronkelijk bevindt de FDC zich op adres #0A00 tot #0A7F. Verschillende mensen verplaatsten de FDC naar het #Bxxx gebied, doch dit leek ons slechts een halve oplossing. Bovenin de DOSROM is nog voldoende ruimte vrij, zelfs na uitbreiding met DECADOS. Bovendien hoeft er in de ATOM niets gewijzigd te worden omdat het #Exxx gebied toch al buiten de bus is gebracht t.b.v. de DOSROM. Enige voorwaarde is het disable'n van de DOSROM op het moment dat de FDC geadresseerd wordt. Om dit te bereiken wordt pin 9 van IC 11 losgemaakt van de NRDS. Pin 9 van IC 11 wordt nu verbonden met pin 12 van IC 4.
Met deze handeling wordt tevens voldaan aan de wijziging in ATOM-NIEUWS 85 no.7 blz.13. Vervolgens wordt IC 22 (LS138) verwijderd en een LS133 wordt in z'n plaats gezet. Dan van IC 11 de pinnen 5 en 6 met 8 verbinden. Pin 5 van de voet van IC 22 losmaken van BLK-0 (componentenzijde). Pin 5 van IC 22 verbinden met pin 4 van IC 11. Pin 14 van IC 22 doorkrassen en pin 13 van IC 4 verbinden met pin 9 van IC 22. De pinnen 13, 14 en 15 van IC 22 verbinden met +5V (pin 16 van IC 22). Tenslotte de pinnen 7, 10, 11 en 12 verbinden met de adreslijnen A3 t/m A6. Het principeschema verduidelijkt e.e.a. Ook deze schakeling kan eenvoudig worden voorzien van een enableoutput. Twee weerstanden, een zenerdiode en en transistor zijn de benodigde onderdelen. Deze onderdelen laat u gewoon weg indien u van de enableoutput geen gebruik maakt.
Vervolgens de softwarematige kant van de zaak.

Aanpassing DOSROM t.b.v. FDC op adres #EFF8 met toevoeging van DECADOS.

De software ziet er als volgt uit:

*1. Laden we DECADOS van Frans v. Hoesel uit AN4.2.85 blz.39 en verwijderen hieruit de regels 220 t/m 240.
*2. Run dit programma waarbij de DOSROM aangepast is met DECADOS van #5000 t/m #5FFF.
*3. Laden de FDC aanpassingsroutine die hierbij gegeven is.
*4. Run dit programma. Geef als high-byte #EF en als low-byte #F8. Zie hiervoor de regels 45 en 55.
*5. Programmeer de nieuwe DOS-EFF8 van #5000 t/m #5FFF in een EPROM.

```
 10 PROGRAM FDC AANPASSINGSROUTINE
 20
 30 REM Acorn Broodje 2.4.blz.54
 40 IN."WAARDE HIGH BYTE WELKE WAS"'"#0A WORDT"A
 45 REM A=#EF
 50 IN."WAARDE LOW  BYTE WELKE WAS"'"#00 WORDT"B
 55 REM B=#F8
 60 P.&A,&B
 70 ?#5004=A
 80 ?#5009=A
 90 ?#5228=A
100 ?#522D=A
110 ?#57DD=A
```



```
120 ?#57E2=A
130 ?#57E6=A
140 ?#57E8=A
150 ?#580C=A
160 ?#5814=A
170 ?#5851=A
180 ?#585F=A
190 ?#587D=A
200 ?#5227=B
210 ?#522C=B
220 ?#57DC=B
230 ?#57E1=B
240 ?#57E5=B
250 ?#580B=B
260 ?#587C=B
270 ?#57EA=B+1
280 ?#5813=B+1
290 ?#5003=B+2
300 ?#5008=B+2
310 ?#5850=B+4
320 ?#585E=B+4
330 P."OK"'
340 END
```

Zie voor het schema pagina 4

N.B. Doordat in de club de wens werd uitgesproken de FDC te verplaatsen naar een standaardadres leek het voor ons duidelijk dat hiervoor maar 1 plaats in aanmerking kwam, n.l. #EFF8-#EFFF. Alles binnen de ruimte van de DOSROM. Zodoende wordt niet onnodig weer beslag gelegd op de toch al schaarse ruimte.
GOED IDEE, MET DE FDC NAAR BLOK E !!!

A. Peters & A. Wessels reg. Arnhem

Wilt u lid worden van de ATOM COMPUTER CLUB.

Neem dan contact op met de penningmeester van de regio waar u bij ingedeeld wilt worden. Hij kan u vertellen, waar de regiobijeenkomsten zijn en wat u als lid kunt verwachten.

Regio NOORD:
D. Uuldriks Wiemers 14 9642 KG Veendam 05987-19611

Regio OVERIJSSEL/GELDERLAND:
N. van Wijnen Korte Dreef 12 8302 BS Emmeloord 05270-13583

Regio TWENTE:
D. ter Harmsel R.v. Schevenstraat 37 7521 SC Enschede 053-359034

Regio NOORD-HOLLAND:
P. van Kuik Zuideinde 54-a 1843 JP Groot-Schermer 02997-1902

Regio DEN HAAG:
Th. Waayer L. Couperusstraat 6 2274 XP Voorburg 070-862504

Regio DELFT:
F. von Morgen M. de Ruyterweg 21 2628 BA Delft

Regio ROTTERDAM:
R. de Haan Brasem 125 2986 HA Ridderkerk 01804-25160

Regio CENTRUM:
P. van Mourik Ruiterstede 60 3431 XN Nieuwegein 03402-48781

Regio ARNHEM:
J. Hartog Keynenbergseweg 60 6871 WK Renkum 08373-13757

Regio ZEELAND:
E. Gijssel Dorpsstraat 86 4424 CZ Wemeldinge 01192-2121

Regio BRABANT-OOST:
P. Ehrlich Roostenlaan 266 5644 BS Eindhoven 040-114183

Regio LIMBURG:
A. v. Zandvoort Mozartstraat 328 6044 RS Roermond 04750-21797

Regio BELGIE:
R. Leyssens Oude Baan 127 3550 Heusden Belgie

Eventueel kunt u de contributie rechtstreeks overmaken per bank of giro aan de Atom Computer Club te Groenekan. Vermeld echter uw volledige naam, adres en de REGIO waarbij u ingedeeld wilt worden.